Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/004676

International filing date: 16 March 2005 (16.03.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-086174
Filing date: 24 March 2004 (24.03.2004)

Date of receipt at the International Bureau: 28 April 2005 (28.04.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 Date of Application:	２００４年　３月２４日
出 願 番 号 Application Number:	特願２００４－０８６１７４
パリ条約による外国への出願に用いる優先権の主張の基礎となる出願の国コードと出願番号 The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is	ＪＰ２００４－０８６１７４
出 願 人 Applicant(s):	松下電器産業株式会社

２００５年　４月１３日

特許庁長官
Commissioner,
Japan Patent Office

小 川

【書類名】　特許願
【整理番号】　2037950002
【あて先】　特許庁長官殿
【国際特許分類】　G06F 12/08　310
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　中西　龍太
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　岡林　はづき
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　田中　哲也
【発明者】
　【住所又は居所】　大阪府門真市大字門真１００６番地　松下電器産業株式会社内
　【氏名】　清原　督三
【特許出願人】
　【識別番号】　000005821
　【氏名又は名称】　松下電器産業株式会社
【代理人】
　【識別番号】　100109210
　【弁理士】
　【氏名又は名称】　新居　広守
【手数料の表示】
　【予納台帳番号】　049515
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　特許請求の範囲　1
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
　【包括委任状番号】　0213583

【書類名】特許請求の範囲
【請求項１】
プロセッサの状態に関する条件を生成する条件生成手段と、
現在のプロセッサの状態が前記条件を満たすかどうかを判定する判定手段と、
操作対象となるアドレスを生成するアドレス生成手段と、
前記判定手段が条件を満たすと判定したときに前記アドレス生成手段によって生成されたアドレスを用いてキャッシュを操作する操作手段と
を備えることを特徴とするキャッシュメモリシステム。
【請求項２】
前記条件生成手段は、前記判定手段が条件を満たすと判定した場合に新たな条件を生成する
ことを特徴とする請求項１記載のキャッシュメモリシステム。
【請求項３】
前記条件生成手段は、プロセッサ内の特定レジスタの値に関する条件を生成する
ことを特徴とする請求項２記載のキャッシュメモリシステム。
【請求項４】
前記特定レジスタはプログラムカウンタである
ことを特徴とする請求項３記載のキャッシュメモリシステム。
【請求項５】
前記条件生成手段は、特定のアドレス範囲内へのメモリアクセスおよび特定のアドレス範囲外へのメモリアクセスの何れかを前記条件として生成する
ことを特徴とする請求項２記載のキャッシュメモリシステム。
【請求項６】
前記条件生成手段は、プロセッサが特定命令を実行することを前記条件として生成する
ことを特徴とする請求項１記載のキャッシュメモリシステム。
【請求項７】
前記条件生成手段は、現在の条件に特定の演算を施すことによって前記新たな条件を生成する
ことを特徴とする請求項２記載のキャッシュメモリシステム。
【請求項８】
前記条件生成手段はメモリアクセスアドレスを条件として生成し、
前記判定手段が条件を満たすと判定した場合に現在の条件に定数を加算することによって前記新たな条件を生成する
ことを特徴とする請求項７記載のキャッシュメモリシステム。
【請求項９】
前記定数は、プロセッサにより実行されるポストインクリメント付きロード／ストア命令におけるインクリメント値またはデクリメント値、およびプロセッサにより実行される２回のロード／ストア命令におけるアドレスの差分値の何れかである
ことを特徴とする請求項８記載のキャッシュメモリシステム。
【請求項１０】
前記条件生成手段は複数の条件を生成し、
前記判定手段は、複数の条件のすべてを満たすかどうかを判定する
ことを特徴とする請求項１記載のキャッシュメモリシステム。
【請求項１１】
前記条件生成手段は複数の条件を生成し、
前記判定手段は、複数の条件の何れかを満たすかどうかを判定する
ことを特徴とする請求項１記載のキャッシュメモリシステム。
【請求項１２】
前記操作手段は、
前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成され

たアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、
格納されていないと判定された場合に、キャッシュメモリ中のラインを選択する選択手段と、
前記選択されたラインが有効でダーディならライトバックを行うライトバック手段と、
前記アドレスに対応するデータをメモリからライトバック後の選択されたラインへ転送する転送手段と、
前記アドレスをタグとして前記選択されたラインへ登録する登録手段と
を備えることを特徴とする請求項１から３の何れかに記載のキャッシュメモリシステム。

【請求項１３】
前記操作手段は、
前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、
格納されていないと判定された場合に、キャッシュメモリ中のラインを選択する選択手段と、
選択されたラインが有効でダーティであれば、ライトバックを行うライトバック手段と、
メモリから選択されたラインへデータを転送することなく、前記生成したアドレスをタグとして選択されたラインへ登録する登録手段と
を備えることを特徴とする請求項１から３の何れかに記載のキャッシュメモリシステム。

【請求項１４】
前記操作手段は、
前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、
格納されていると判定された場合に、キャッシュメモリ中の格納先のラインを選択する選択手段と、
選択されたラインが有効でかつダーディであればライトバックを行うライトバック手段と、
を備えることを特徴とする請求項１から３の何れかに記載のキャッシュメモリシステム。

【請求項１５】
前記操作手段は、
前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、
格納されていると判定された場合に、キャッシュメモリ中の格納先のラインを選択する選択手段と、
選択されたラインを無効化する無効化手段と
を備えることを特徴とする請求項１から３の何れかに記載のキャッシュメモリシステム。

【請求項１６】
前記操作手段は、
前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、
格納されていると判定された場合に、キャッシュメモリ中の格納先のラインを選択する

選択手段と、
ラインのアクセス順序を示す順序情報に対して、選択されたラインのアクセス順序を変更する変更手段と、
を備えることを特徴とする請求項１から３の何れかに記載のキャッシュメモリシステム。
【請求項１７】
前記条件生成手段により前記条件としてメモリアドレスを生成し、
前記操作手段は、さらに、
前記条件生成手段により生成されたメモリアドレスがラインの途中を指す場合に、当該ラインの先頭、次のラインの先頭および前のラインの先頭の何れかを指すように調整することによりアドレスを生成する調整手段を備える
ことを特徴とする請求項１２から１６の何れかに記載ののキャッシュシステム。
【請求項１８】
キャッシュメモリの制御方法であって、
プロセッサの状態に関する条件を生成する条件生成ステップと、
現在のプロセッサの状態が前記条件を満たすかどうかを判定する判定ステップと、
操作対象となるアドレスを生成するアドレス生成ステップと、
前記判定ステップにおいて条件を満たすと判定したときに前記アドレス生成ステップにおいて生成されたアドレスを用いてキャッシュを操作する操作ステップと
を有することを特徴とする制御方法。

【書類名】明細書
【発明の名称】キャッシュメモリ及びその制御方法
【技術分野】
【０００１】
本発明は、プロセッサのメモリアクセスを高速化するためのキャッシュメモリ及びその制御方法に関する。
【背景技術】
【０００２】
近年のマイクロプロセッサでは、例えば、ＳＲＡＭ（Ｓｔａｔｉｃ Ｒａｎｄｏｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）等から成る小容量で高速なキャッシュメモリをマイクロプロセッサの内部、もしくはその近傍に配置し、データの一部をキャッシュメモリに記憶することによって、マイクロプロセッサのメモリアクセスを高速化させている。
【０００３】
キャッシュの効率向上(ヒット率向上、キャッシュミスレイテンシ低減)のため、キャッシュミスが発生する前に、近い未来に使用するデータを予めキャッシュにフィルする技術としてプリロード（又はプリフェッチ)がある（例えば特許文献１）。
【０００４】
従来のプリフェッチ技術では、プリフェッチ命令により指定したアドレスを含むラインをキャッシュにロードしている。これにより、キャッシュミスの低減を図っている。
【特許文献１】特開平７－２９５８８２号公報
【発明の開示】
【発明が解決しようとする課題】
【０００５】
しかしながら、上記従来技術によれば、 例えば、ソフトウェアによって、ループの外側で一括してプリフェッチを行った場合、ループで必要なデータ領域をあらかじめすべてキャッシュ上に確保することになるため、キャッシュの容量が小さい場合には、それ以外の必要なデータがキャッシュから追い出され、キャッシュミスが発生する。また、データのキャッシング終了、無効化等をループの外側で一括して行う場合は、ループを抜けるまでこれらの動作によるキャッシュの開放が行われないため、キャッシュの容量が不足し、キャッシュミスが発生する。
【０００６】
また、ループ中にソフトウェアによってキャッシュ操作を行う命令を挿入した場合、キャッシュ操作を行うアドレスをループの中でソフトウェアによって管理する必要がある。つまり、キャッシュ操作を行う命令をループの中に記述する必要があるため、性能の劣化が発生する。
【０００７】
さらに、メモリへのアクセスの状況をハードウェアによって監視し、ハードウェアによって自動的にキャッシュ操作をする場合、正確な予測が行えないと無駄な転送が発生する、あるいは、ソフトウェアからの正確な情報がないとキャッシュ上のデータと外部メモリ上のデータの整合性が取ることができないため、ハードウェアの予測によってこれらの操作を行うことは困難である。
【課題を解決するための手段】
【０００８】
上記課題を解決するため本発明のキャッシュメモリシステムは、プロセッサの状態に関する条件を生成する条件生成手段と、現在のプロセッサの状態が前記条件を満たすかどうかを判定する判定手段と、操作対象となるアドレスを生成するアドレス生成手段と、前記判定手段が条件を満たすと判定したときに前記アドレス生成手段によって生成されたアドレスを用いてキャッシュを操作する操作手段とを備える。
【０００９】
ここで、前記条件生成手段は、前記判定手段が条件を満たすと判定した場合に新たな条

件を生成するようにしてもよい。
この構成によれば、プロセッサの動作状態が条件を満たすようになったときにキャッシュを操作するので、プロセッサの動作の進行状況に同期してキャッシュメモリを操作することができる。またソフトウェア的に実現しないのでプロセッサに負荷をかけることなくキャッシュメモリを効率よく性能劣化を招くことなく動作させることができる。
【００１０】
ここで、前記条件生成手段は、プロセッサ内の特定レジスタの値に関する条件を生成するようにしてもよい。前記特定レジスタはプログラムカウンタであってもよい。
この構成によれば、メモリアクセスアドレスや、プログラムフェッチアドレスを前記条件として、プロセッサの状態を監視することができる。
【００１１】
ここで、前記条件生成手段は、特定のアドレス範囲内へのメモリアクセスおよび特定のアドレス範囲外へのメモリアクセスの何れかを前記条件として生成するようにしてもよい。
【００１２】
また、前記条件生成手段は、プロセッサが特定命令を実行することを前記条件として生成するようにしてもよい。
ここで、前記条件生成手段は、現在の条件に特定の演算を施すことによって前記新たな条件を生成するようにしてもよい。
【００１３】
また、前記条件生成手段はメモリアクセスアドレスを条件として生成し、前記判定手段が条件を満たすと判定した場合に現在の条件に定数を加算することによって前記新たな条件を生成するようにしてもよい。
【００１４】
ここで、前記定数は、プロセッサにより実行されるポストインクリメント付きロード／ストア命令におけるインクリメント値またはデクリメント値、およびプロセッサにより実行される２回のロード／ストア命令におけるアドレスの差分値の何れかであるようにしてもよい。
【００１５】
ここで、前記条件生成手段は複数の条件を生成し、前記判定手段は、複数の条件のすべてを満たすかどうかを判定するようにしてもよい。
また、前記条件生成手段は複数の条件を生成し、前記判定手段は、複数の条件の何れかを満たすかどうかを判定するようにしてもよい。
【００１６】
ここで、前記操作手段は、前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、格納されていないと判定された場合に、キャッシュメモリ中のラインを選択する選択手段と、前記選択されたラインが有効でダーディならライトバックを行うライトバック手段と、前記アドレスに対応するデータをメモリからライトバック後の選択されたラインへ転送する転送手段と、前記アドレスをタグとして前記選択されたラインへ登録する登録手段とを備えるようこうせいしてもよい。
【００１７】
この構成によれば、キャッシュメモリのプリフェッチを、プロセッサの動作状況を監視して同期を取りつつ適切なタイミングで行うことができる。
ここで、前記操作手段は、前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、格納されていないと判定された場合に、キャッシュメモリ中のラインを選択する選択手段と、選択されたラインが有効でダーティであれば、ライトバックを行うライトバック手段と、メモリから選択されたラインへデータを転送することなく、前記生成したアドレスをタグとして選択されたラインへ登録する登録手段と

を備える構成としてもよい。
【００１８】
この構成によれば、キャッシュメモリのラインにデータを転送するとなく確保すること（タッチと呼ぶ）を、プロセッサの動作状況を監視して同期を取りつつ適切なタイミングで行うことができる。
【００１９】
ここで、前記操作手段は、前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、格納されていると判定された場合に、キャッシュメモリ中の格納先のラインを選択する選択手段と、選択されたラインが有効でかつダーディであればライトバックを行うライトバック手段とを備える構成としてもよい。
【００２０】
この構成によれば、キャッシュメモリのラインデータのライトバックを（キャッシング終了）を、プロセッサの動作状況を監視して同期を取りつつ適切なタイミングで行うことができる。
【００２１】
ここで、前記操作手段は、前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、格納されていると判定された場合に、キャッシュメモリ中の格納先のラインを選択する選択手段と、選択されたラインを無効化する無効化手段とを備える構成としてもよい。
【００２２】
この構成によれば、キャッシュメモリのラインの無効化を、プロセッサの動作状況を監視して同期を取りつつ適切なタイミングで行うことができる。
ここで、前記操作手段は、前記判定手段が条件を満たすと判定したときに、前記アドレス生成手段により生成されたアドレスに対応するデータがキャッシュに格納されているかどうかを判定するデータ判定手段と、格納されていると判定された場合に、キャッシュメモリ中の格納先のラインを選択する選択手段と、ラインのアクセス順序を示す順序情報に対して、選択されたラインのアクセス順序を変更する変更手段とを備える構成としてもよい。
【００２３】
この構成によれば、キャッシュメモリのラインのアクセス順序情報を、プロセッサの動作状況を監視して同期を取りつつ適切なタイミングで行うことができる。これによりいわゆるＬＲＵによるキャッシュのリプレース順序を早めたり遅くしたりすることができる。
【００２４】
ここで、前記条件生成手段により前記条件としてメモリアドレスを生成し、前記操作手段は、さらに、前記条件生成手段により生成されたメモリアドレスがラインの途中を指す場合に、当該ラインの先頭、次のラインの先頭および前のラインの先頭の何れかを指すように調整することによりアドレスを生成する調整手段を備える構成としてもよい。
【００２５】
この構成によれば、プロセッサがアドレス昇順にアクセスする場合も、アドレス高順にアクセスする場合も、次に必要なラインのアドレスを適切に算出することができる。
また、本発明のキャッシュメモリの制御方法についても上記と同様の手段、作用を有する。
【発明の効果】
【００２６】
本発明のキャッシュメモリによれば、プロセッサの動作の進行状況に同期してキャッシュメモリを操作することができる。またソフトウェア的に実現しないのでプロセッサに負荷をかけることなくキャッシュメモリを効率よく性能劣化を招くことなく動作させることができる。

【００２７】
例えば、プロセッサがポストインクリメント付きロード命令によってシーケンシャルにメモリアクセスする場合には、効率よくプリフェッチすることができる。また、プロセッサがポストインクリメント付きストア命令によってシーケンシャルにデータを書き込む場合には、メモリからキャッシュメモリにデータをロードするペナルティを削除することができ、さらに効率よくキャッシュエントリーをタッチ（確保）することができる。
【００２８】
また、プロセッサがポストインクリメント付きストア命令によってシーケンシャルにデータを書き込む場合に、プロセッサのストア命令の進行状況に応じて、ストアが完了したラインを予測し、予測したラインにキャッシング終了属性や無効化属性を設定するので、プロセッサでは、キャッシュメモリのラインサイズやライン境界を認識しなくても、ライトバックすべきラインをキャッシュメモリにおいて予測し、効率よくクリーニング（ライトバック）や、キャシュエントリーの開放（無効化）を行うことができる。
【００２９】
さらに、プロセッサがポストインクリメント付きロード命令によってシーケンシャルにデータを読み出す場合に、読み出しが完了したラインデータを保持するキャッシュエントリーにリプレース属性が設定され、真っ先にリプレース対象として選択されるので、アクセス頻度の低いデータがキャッシュメモリに残ることによるキャッシュミスの誘発を低減することができる。
【発明を実施するための最良の形態】
【００３０】
（実施の形態１）
＜全体構成＞
図１は、本発明の実施の形態１におけるプロセッサ１、キャッシュメモリ３、メモリ２を含むシステムの概略構成を示すブロック図である。同図のように、本発明のキャッシュメモリ３は、プロセッサ１およびメモリ２を有するシステムに備えられる。
【００３１】
キャッシュメモリ３は、プロセッサ１から指定された条件に従って、予測プリフェッチを行うよう構成されている。予測プリフェッチとは、プロセッサ１によるメモリアクセスの進行状況に基づいて次にプリフェッチすべきラインアドレスを予測し、予測したラインアドレスのデータをキャッシュメモリにプリフェッチすることをいう。また、プロセッサ１から指定される条件というのは、アドレス範囲、シーケンシャルアクセスする場合のアドレスのインクリメント値またはデクリメント値、またはアクセス方向（アドレス昇順または降順）等である。予測プリフェッチは、プロセッサがシーケンシャルにメモリアクセスする場合に適している。
【００３２】
加えて、キャッシュメモリ３は、プロセッサ１から指定された条件に従って、予測タッチを行うよう構成されている。予測タッチとは、プロセッサ１によるメモリアクセスの進行状況に基づいて次にタッチすべきラインアドレスを予測し、予測したラインアドレスのデータをロードすることなくセットキャッシュエントリーにバリッドフラグとタグとを設定することにより確保することをいう。予測タッチは、配列データなどの演算結果をメモリに順次格納する場合などに適している。
【００３３】
＜キャッシュメモリの構成＞
以下、キャッシュメモリ３の具体例として、４ウェイ・セット・アソシエイティブ方式のキャッシュメモリに本発明を適用した場合の構成について説明する。
【００３４】
図２は、キャッシュメモリ３の構成例を示すブロック図である。同図のように、キャッシュメモリ３は、アドレスレジスタ２０、メモリＩ／Ｆ２１、デコーダ３０、４つのウェイ３１ａ～３１ｄ（以下ウェイ０～３と略す）、４つの比較器３２ａ～３２ｄ、４つのア

ンド回路３３ａ～３３ｄ、オア回路３４、セレクタ３５、３６、デマルチプレクサ３７、制御部３８を備える。

【００３５】

アドレスレジスタ２０は、メモリ２へのアクセスアドレスを保持するレジスタである。このアクセスアドレスは３２ビットであるものとする。同図に示すように、アクセスアドレスは、最上位ビットから順に、２１ビットのタグアドレス、４ビットのセットインデックス（図中のＳＩ）、５ビットのワードインデックス（図中のＷＩ）を含む。ここで、タグアドレスはウェイにマッピングされるメモリ中の領域（そのサイズはセット数×ブロックである）を指す。この領域のサイズは、タグアドレスよりも下位のアドレスビット（Ａ１０～Ａ０）で定まるサイズつまり２ｋバイトであり、１つのウェイのサイズでもある。セットインデックス（ＳＩ）はウェイ０～３に跨る複数セットの１つを指す。このセット数は、セットインデックスが４ビットなので１６セットある。タグアドレスおよびセットインデックスで特定されるキャッシュエントリーは、リプレース単位であり、キャッシュメモリに格納されている場合はラインデータ又はラインと呼ばれる。ラインデータのサイズは、セットインデックスよりも下位のアドレスビットで定まるサイズつまり１２８バイトである。１ワードを４バイトとすると、１ラインデータは３２ワードである。ワードインデックス（ＷＩ）は、ラインデータを構成する複数ワード中の１ワードを指す。アドレスレジスタ２０中の最下位２ビット（Ａ１、Ａ０）は、ワードアクセス時には無視される。

【００３６】

メモリＩ／Ｆ２１は、キャッシュメモリ３からメモリ２へのデータのライトバックや、メモリ２からキャッシュメモリ３へのデータのロード等、キャッシュメモリ３からメモリ２をアクセスするためのＩ／Ｆである。

【００３７】

デコーダ３０は、セットインデックスの４ビットをデコードし、４つのウェイ０～３に跨る１６セット中の１つを選択する。

４つのウェイ０～３は、同じ構成を有数する４つのウェイであり、４×２ｋバイトの容量を有する。各ウェイは、１６個のキャッシュエントリーを有する。１つのキャッシュエントリーは、バリッドフラグＶ、２１ビットのタグ、１２８バイトのラインデータ、ダーティフラグＤを有する。タグは２１ビットのタグアドレスのコピーである。ラインデータは、タグアドレスおよびセットインデックスにより特定されるブロック中の１２８バイトデータのコピーである。バリッドフラグＶは、当該キャッシュエントリーのデータが有効か否かを示す。ダーティフラグＤは、当該キャッシュエントリーにプロセッサから書き込みがあったか否か、つまりサブライン中にキャッシュされたデータが存在するが書き込みによりメモリ中のデータと異なるためメモリに書き戻すことが必要か否かを示す。

【００３８】

比較器３２ａは、アドレスレジスタ２０中のタグアドレスと、セットインデックスにより選択されたセットに含まれる４つのタグ中のウェイ０のタグとが一致するか否かを比較する。比較器３２ｂ～３２ｃについても、ウェイ３１ｂ～３１ｄに対応すること以外は同様である。

【００３９】

アンド回路３３ａは、バリッドフラグと比較器３２ａの比較結果とが一致するか否かを比較する。この比較結果をｈ０とする。比較結果ｈ０が１である場合は、アドレスレジスタ２０中のタグアドレスおよびセットインデックスに対応するラインデータが存在すること、つまりウェイ０においてヒットしたことを意味する。比較結果ｈ０が０である場合は、ミスヒットしたことを意味する。アンド回路３３ｂ～３３ｄについても、ウェイ３１ｂ～３１ｄに対応すること以外は同様である。その比較結果ｈ１～ｈ３は、ウェイ１～３でヒットしたかミスしたかを意味する。

【００４０】

オア回路３４は、比較結果ｈ０～ｈ３のオアをとる。このオアの結果をｈｉｔとする。

ｈｉｔは、キャッシュメモリにヒットしたか否かを示す。
セレクタ３５は、選択されたセットにおけるウェイ０～３のラインデータのうち、ヒットしたウェイのラインデータを選択する。
【００４１】
セレクタ３６は、セレクタ３５により選択された３２ワードのラインデータにうち、ワードインデックスに示される１ワードを選択する。
デマルチプレクサ３７は、キャッシュエントリーにデータを書き込む際に、ウェイ０～３の１つに書き込みデータを出力する。この書き込みデータはワード単位でよい。
【００４２】
制御部３８は、予測処理部３９を含み、キャッシュメモリ３の全体の制御を行う。予測処理部３９は、主としてプリフェッチの制御を行う。
【００４３】
＜予測処理部の構成＞
図３は、予測処理部３９の構成例を示すブロック図である。同図のように予測処理部３９は、コマンドレジスタ４０１、スタートアドレスレジスタ４０２、サイズレジスタ４０３、加算器４０４、スタートアライナ４０５ａ、４０５ｂ、エンドアライナ４０６ａ、４０６ｂ、アクセスアドレスレジスタ４０７、予測値レジスタ４０８、定数レジスタ４０９、セレクタ４１０、加算器４１１、比較器４１２、実行部４１３を備える。
【００４４】
コマンドレジスタ４０１は、プロセッサ１から直接アクセス可能なレジスタであり、プロセッサ１により書き込まれたコマンドを保持する。図４（ｃ）に、コマンドレジスタ４０１にコマンドを書き込む命令の一例を示す。この命令は、通常の転送命令（ｍｏｖ命令）であり、ソースオペランドとしてコマンドを、デスティネーションオペランドとしてコマンドレジスタ（ＣＲ）４０１を指定している。図４（ｄ）に、コマンドフォーマットの一例を示す。このコマンドフォーマットは、コマンド内容と定数とからなる。ここで、コマンド内容は、予測プリフェッチコマンドと、予測タッチコマンドとの何れかを表す。定数は、プロセッサ１が、シーケンシャルにメモリアクセスする場合のメモリアクセスアドレスのインクリメント値またはデクリメント値を表し、例えば、＋４、＋８、－４、－８等である。なお、コマンド中の定数の代わりに、アクセス方向（アドレス昇順かアドレス降順か）と絶対値（連続アクセスする場合のアドレスの差分）とを含むようにしてもよい。
【００４５】
スタートアドレスレジスタ４０２は、プロセッサ１から直接アクセス可能なレジスタであり、プロセッサ１により書き込まれたスタートアドレスを保持する。このスタートアドレスはＣフラグ（キャッシングを終了してよいか否かを示すクリーニングフラグ）を設定すべきアドレス範囲の開始位置を示す。図４（ａ）に、スタートアドレスレジスタ４０２にスタートアドレスを書き込む命令の一例を示す。この命令も、図４（ｃ）と同様に通常の転送命令（ｍｏｖ命令）である。
【００４６】
サイズレジスタ４０３は、プロセッサ１から直接アクセス可能なレジスタであり、プロセッサ１により書き込まれたサイズを保持する。このサイズは、スタートアドレスからのアドレス範囲を示す。図４（ｂ）に、サイズレジスタ４０３にサイズを書き込む命令の一例を示す。この命令も、図４（ｃ）と同様に通常の転送命令（ｍｏｖ命令）である。なお、サイズの単位は、バイト数であっても、ライン数（キャッシュエントリー数）であってもよく、予め定められた単位であればよい。
【００４７】
加算器４０４は、スタートアドレスレジスタ４０２に保持されたスタートアドレスとサイズレジスタ４０３に保持されたサイズとを加算する。加算結果は、アドレス範囲の終了位置を指すエンドアドレスである。加算器４０４は、サイズがバイト数指定の場合はバイトアドレスとして加算し、サイズがライン数指定の場合はラインアドレスとして加算すれ

ばよい。
【００４８】
スタートアライナ４０５ａ、４０５ｂは、スタートアドレスをライン境界の位置に調整する。スタートアライナ４０５ａはエンドアドレスの方向に、４０５ｂはエンドアドレスとは反対の方向に調整する。この調整によりプロセッサ１はラインサイズ及びライン境界とは無関係に任意のアドレスをスタートアドレスとして指定することができる。
【００４９】
エンドアライナ４０６ａ、４０６ｂは、エンドアドレスをライン境界の位置に調整する。エンドアライナ４０６５ａはスタートアドレスの方向に、４０６ｂはステートアドレスとは反対の方向に調整する。この調整によりプロセッサ１はラインサイズ及びライン境界とは無関係に任意の大きさを上記サイズとして指定することができる。
【００５０】
図５に、スタートアライナ４０５ａ、４０５ｂ及びエンドアライナ４０６ａ、４０６ｂの説明図を示す。同図において、プロセッサ１から指定されたスタートアドレスはラインＮの途中の任意の位置を指す。スタートアライナ４０５ａは、次のライン（Ｎ＋１）の先頭を指すよう調整し、調整後のアドレスをアラインスタートアドレスａとして出力する。スタートアライナ４０５ｂは、スタートアドレスのデータを含むラインＮの先頭を指すよう調整し、調整後のアドレスをアラインスタートアドレスｂとして出力する。アラインスタートアドレスが指すラインをスタートラインと呼ぶ。
【００５１】
また、エンドアドレスはラインＭの途中の任意の位置を指す。エンドアライナ４０６ａは、直前のライン（Ｍ－１）の先頭を指すよう調整し、調整後のアドレスをアラインエンドアドレスａとして出力する。エンドアライナ４０６ｂは、エンドアドレスのデータを含むラインＭの先頭を指すよう調整し、調整後のアドレスをアラインエンドアドレスｂとして出力する。アラインエンドアドレスが指すラインをエンドラインと呼ぶ。
【００５２】
同図のように、スタートアライナ４０５ａ及びエンドアライナ４０６ａはライン単位で内側アラインを行う。スタートアライナ４０５ｂ及びエンドアライナ４０６ｂはライン単位で外側アラインを行う。さらに、ライン単位の外側アラインの後、さらに、サブライン単位の外側アラインと内側アラインが可能である。
【００５３】
アクセスアドレスレジスタ４０７は、プロセッサ１からのメモリアクセスアドレスを保持する。
予測値レジスタ４０８は、初期値として、アクセスアドレスレジスタ４０７のメモリアクセスアドレスと定数レジスタ４０９の定数とを加算した値を予測値として保持し、以降、プロセッサ１がメモリアクセスを実行したときに、アクセスアドレスレジスタ４０７のメモリアクセスアドレスと予測値とが一致していれば、予測値と定数レジスタ４０９の定数とを加算した値を新たな予測値として更新し、一致していなければ、新たな初期値に更新する。
【００５４】
定数レジスタ４０９は、プロセッサ１によるメモリアクセスアドレスのインクリメント値またはデクリメント値を保持する。このインクリメント値（またはデクリメント値）は、プロセッサ１がメモリをシーケンシャルにアクセスする場合のポストインクリメント付きロード／ストア命令のインクリメント値（またはデクリメント値）であり、例えば、図４（ｄ）に示したコマンド中の定数として指定される。
【００５５】
セレクタ４１０は、アクセスアドレスレジスタ４０７のメモリアクセスアドレスと予測値レジスタ４０８の予測値とが一致していれば、予測値レジスタ４０８を選択し、一致していなければ、アクセスアドレスレジスタ４０７を選択する。
【００５６】

加算器４１１は、セレクタ４１０に選択された予測値またはメモリアクセスアドレスと、定数レジスタ４０９の定数とを加算する。加算後の値は、新たな予測値又は新たな初期値として予測値レジスタ４０８に保持される。

【００５７】

比較器４１２は、アクセスアドレスレジスタ４０７のメモリアクセスアドレスと予測値レジスタ４０８の予測値とが一致するか否かを判定する。

実行部４１３は、プリフェッチ部４１４とタッチ部４１５とを備える。

【００５８】

プリフェッチ部４１４は、プロセッサのロード命令の進行状況に応じて次にロードされるライン推定し、推定したラインをプリフェッチする予測プリフェッチを行う。

タッチ部４１５は、プロセッサのストア命令の進行状況に応じて次にストアされるライン推定し、推定したラインを保持するためのキャッシュエントリーを、メモリからデータをロードすることなく確保する（Ｖ＝１の設定とタグの設定とを行う）。

【００５９】

＜予測プリフェッチ＞

図６（ａ）は、プリフェッチ部４１４による予測プリフェッチの説明図である。同図（ａ）において、ラインＮ、Ｎ＋１、・・・、Ｎ＋ｎはアドレスが連続するラインを示す。スタートアドレスレジスタ４０２及びサイズレジスタ４０３により定まるスタートライン、エンドラインは、それぞれラインＮ＋１、ラインＮ＋ｎであるものとする。また、ＬＤ（１）、ＬＤ（２）、・・・は、ロード命令によるアクセス位置を示している。

【００６０】

プリフェッチ部４１４は、最初のロード命令ＬＤ（１）の実行時に、そのメモリアクセスアドレスに定数を加算した値を初期値として予測値レジスタ４０８に保持させる。２回目のロード命令ＬＤ（２）の実行時に、そのメモリアクセスアドレスと予測値レジスタ４０８の予測値とが一致した場合、プリフェッチ部４１４は、ポストインクリメント付きロード命令によるシーケンシャルアクセスであるものと推定して、次のラインＮ＋１をプリフェッチする。もし、次のラインがキャッシュメモリに格納されている場合は、なにもしない。

【００６１】

プロセッサ１によってポストインクリメント付きロード命令によりシーケンシャルアクセスがなされた場合、予測値はメモリアクセス毎に一致していくことになる。

このようにして、予測値レジスタ４０８の予測値がラインＮからラインＮ＋１のアドレスにまで更新され、メモリアクセスアドレスと一致したとき、プリフェッチ部４１４は、ラインＮ＋２をプリフェッチする。プリフェッチ部４１４は、この予測プリフェッチをスタートラインからエンドラインまで実行し、エンドラインのプリフェッチが完了すると予測プロフェッチを終了する。

【００６２】

＜予測タッチ＞

図６（ｂ）は、タッチ部４１５による予測タッチの説明図である。図６（ａ）と比較して、プロセッサ１がストア命令によりシーケンシャルにアクセスする場合に、そのメモリアクセスアドレスと予測値レジスタ４０８の予測値とが一致した場合に、ポストインクリメント付きストア命令によるシーケンシャルアクセスと推定し、次のラインＮ＋１に対応するキャッシュエントリーを確保する点が異なる。つまり、プリフェッチの代わりに、キャッシュエントリーにメモリデータをロードしないでタグ及びバリッドフラグを１に設定する点が異なっている。これ以外は予測プリフェッチと同様なので説明を省略する。

【００６３】

＜予測プリフェッチ処理＞

図７は、プリフェッチ部４１４における予測プリフェッチ処理の一例を示すフローチャートである。

【００６４】

同図において、プリフェッチ部４１４は、コマンドレジスタ４０１に予測プリフェッチコマンドが保持されていて（Ｓ４１）、プロセッサがロード命令を実行したとき（Ｓ４２）、当該メモリアクセスアドレスと定数とを加算した値を初期値として予測値レジスタ４０８に設定する（Ｓ４３）。この最初のロード命令のメモリアクセスアドレスと予測値レジスタの予測値とは当然一致しない。予測値レジスタ４０８はクリアされているかランダムな値を保持しているからである。

【００６５】

さらに、プロセッサがロード命令を実行し（Ｓ４４）、かつメモリアクセスアドレスと予測値レジスタの予測値とが一致する場合には（Ｓ４５）、プリフェッチ部４１４は、次ラインのラインアドレスを算出し（Ｓ４６）、算出したラインアドレスがスタートラインからエンドラインまでのアドレス範囲に属していて（Ｓ４７）、次ラインがキャッシュメモリにエントリーされていなければ（Ｓ４８）、次ラインをプリフェッチする（Ｓ４９）。

【００６６】

このプリフェッチにおいて、プリフェッチ部４１４は、ＬＲＵ方式でリプレース対象のウェイを選択し（Ｓ４０１）、当該ウェイのキャッシュエントリーがダーティであればライトバックし（Ｓ４０２、Ｓ４０３）、次ラインのデータを当該キャッシュエントリーにリフィル（プリフェッチ）する（Ｓ４０４）。

【００６７】

さらに、プリフェッチ部４１４は、プリフェッチしたラインがエンドラインであれば（Ｓ５０）予測プリフェッチ処理を終了する。

また、上記Ｓ４５において、メモリアクセスアドレスと予測値レジスタの予測値とが一致しない場合には、メモリアクセスアドレスに定数を加算した値が新たな予測値として予測値レジスタ４０８に設定される。また、一致しない場合は、なにもしない（メモリアクセス待ちの状態）。

【００６８】

このように、プリフェッチ部４１４は、予測プリフェッチにおいて、プロセッサのロード命令の進行状況に応じて次のラインを推定し、推定したラインをプリフェッチするので、プロセッサ１では、キャッシュメモリのラインサイズやライン境界を認識する必要がなく、つまり、プリフェッチ用のアドレスをプロセッサ１にて管理する必要がない。キャッシュメモリにおいて推定に基づいて効率よくプリフェッチを行うことができる。

【００６９】

＜予測タッチ処理＞

図８は、タッチ部４１５における予測タッチ処理の一例を示すフローチャートである。同図は、図７の予測プリフェッチ処理と比較して、Ｓ４１、Ｓ４２、Ｓ４４、Ｓ４９、Ｓ４０４の各ステップの代わりにＳ４１ａ、Ｓ４２ａ、Ｓ４４ａ、Ｓ４９ａ、Ｓ４０４ａを有する点が異なっている。これ以外は同じなので異なる点について説明する。

【００７０】

タッチ部４１５は、Ｓ４１ａにおいて、予測タッチコマンドがコマンドレジスタ４０１に保持されているかを判定し、Ｓ４２ａ、Ｓ４４ａにおいてストア命令が実行されたか否かを判定する。また、タッチ部４１５は、Ｓ４９ａにおいてプリフェッチする代わりにタッチする。すなわち、Ｓ４０４ａにおいて、リプレース対象のキャッシュエントリーメモリデータをロードしないでタグ及びバリッドフラグを１に設定する。

【００７１】

このように、タッチ部４１５は、プロセッサのストア命令の進行状況に応じて次のラインを推定し、推定したラインをタッチする（確保する）ので、プロセッサ１では、キャッシュメモリのラインサイズやライン境界を認識する必要がなく、つまりプリフェッチ用のアドレスをプロセッサ１にて管理する必要がない。キャッシュメモリにおいて推定に基づいて効率よくキャッシュエントリーをタッチすることができる。

【００７２】

＜変形例＞
なお、本発明のキャッシュメモリは、上記の実施の形態の構成に限るものではなく、種々の変形が可能である。以下、変形例のいくつかについて説明する。
（１）定数レジスタ４０９、コマンドにより設定される構成を示したが、（ａ）デフォルトとしてインクリメント値またはデクリメント値を保持する、（ｂ）複数回のロード／ストア命令における２つのメモリアクセスアドレスの差分を算出し、インクリメント値またはデクリメント値として保持する、（ｃ）プロセッサにより指定されたアドレス方向（アドレス昇順かアドレス降順か）に応じて上記（ｂ）を算出し保持する構成としてもよい。
（２）プリフェッチ部４１４において、次のラインのさらに次のライン等複数ラインをプリフェッチするようにしてもよい。同様にタッチ部４１５も、複数ラインをタッチするようにしてもよい。
（３）プリフェッチ部４１４は、比較器４１２で複数回一致した場合にプリフェッチを開始するようにしてもよい。タッチ部４１５も同様である。
（４）上記実施の形態では、４ウェイ・セット・アソシエイティブのキャッシュメモリを例に説明したが、ウェイ数は、いくつでもよい。また、上記実施の形態では、セット数が１６である例を説明したが、セット数はいくつでもよい。
（５）上記実施の形態では、セット・アソシエイティブのキャッシュメモリを例に説明したが、フル・アソシエイティブ方式やダイレクトマップ方式のキャッシュメモリであってもよい。
（６）上記実施の形態では、サブラインのサイズをラインのサイズの１／４としているが、１／２、１／８、１／１６等他のサイズでもよい。その場合、各キャッシュエントリーは、サブラインと同数のバリッドフラグおよびダーティフラグをそれぞれ保持すればよい。
（７）上記実施の形態におけるサイズレジスタ４０３は、サイズを保持する代わりにプリフェッチすべき回数を保持する構成としてもよい。この場合には、予測処理部３９は実際にプリフェッチした回数をカウントし、カウント値がプリフェッチすべき回数に達したとき、プリフェッチを停止する構成とすればよい。このようにプリフェッチ回数をカウントすることにより、プロセッサにおけるプログラム進行状態を監視することができる。
【００７３】
（実施の形態２）
第１の実施の形態では、説明した予測プリフェッチ及び予測タッチでは今後アクセスするであろうラインを推定する場合の構成を説明した。本実施の形態では、既にアクセスしたであろうラインを推定して予測クリーニング（ライトバック）や予測ウィーク化（アクセス順位の最弱化）や予測無効化も行う構成について説明する。
【００７４】
＜キャッシュメモリの構成＞
図９は、本発明の実施の形態２におけるキャッシュメモリの構成を示すブロック図である。同図のキャッシュメモリは、図２に示した構成と比較して、ウェイ３１ａ～３１ｄの代わりにウェイ１３１ａ～１３１ｄを備える点と、制御部３８の代わりに制御部１３８を備える点とが異なっている。以下、同じ点は説明を省略して、異なる点を中心に説明する。
【００７５】
ウェイ１３１ａは、ウェイ３１ａと比べて、各キャッシュエントリー中に、Ｃフラグ、Ｗフラグ及びＵフラグが追加されている点と、ライン単位のバリッドフラグＶの代わりにサブライン毎のバリッドフラグＶ０～Ｖ３を有する点と、ライン単位のダーティフラグＤの代わりにサブライン毎のダーティフラグＤ０～Ｄ３を有する点とが異なる。ウェイ１３１ｂ～１３１ｄも同様である。
【００７６】
図１０に、キャッシュエントリーのビット構成を示す。１つのキャッシュエントリーは、バリッドフラグＶ０～Ｖ３、２１ビットのタグ、１２８バイトのラインデータ、ウィー

クフラグW、使用フラグU及びダーティフラグD０～D３を保持する。
【００７７】
タグは２１ビットのタグアドレスのコピーである。
ラインデータは、タグアドレスおよびセットインデックスにより特定されるブロック中の１２８バイトデータのコピーであり、３２バイトの４つのサブラインからなる。
【００７８】
バリッドフラグＶ０～Ｖ３は、４つのサブラインに対応し、サブラインが有効か否かを示す。
Ｃフラグ（クリーニングフラグ）は、キャッシングを終了してよいかどうかを示すキャッシング終了属性を示す。Ｃ＝０は、以降に書き込みがなされる可能性があることを意味する。Ｃ＝１は、以降に書き込みがなされないことを意味し、ダーティであればクリーニング（ライトバック）によりキャッシングを終了すべきであることを意味する。
【００７９】
ウィークフラグWは、プロセッサからのアクセスに関しては、これ以上使用するか否かを示し、キャッシュメモリにおけるリプレース制御に関しては、他のキャッシュエントリーよりも先に追い出してもよい最弱のリプレース対象であることを示す。
【００８０】
使用フラグUは、そのキャッシュエントリーにアクセスがあったか否かを示し、ＬＲＵ方式におけるキャッシュエントリー間のアクセス順序データの代わりに用いられる。より正確には、使用フラグUの１は、アクセスがあったことを、０はないことを意味する。ただし、１つのセット内の４つウェイの使用フラグが全て１になれば、０にリセットされる。別言すれば、使用フラグUは、アクセスされた時期が古いか新しいか２つの相対的な状態を示す。つまり、使用フラグUが１のキャッシュエントリーは、使用フラグが０のキャッシュエントリーよりも新しくアクセスされたことを意味する。
【００８１】
ダーティフラグD０～D３は、４つのサブラインに対応し、そのサブラインにプロセッサから書き込みがあったか否か、つまりサブライン中にキャッシュされたデータが存在するが書き込みによりメモリ中のデータと異なるためメモリに書き戻すことが必要か否かを示す。
【００８２】
制御部１３８は、制御部３８と比べて、予測処理部３９の代わりに予測処理部１３９を有する点が異なり、予測によりCフラグを設定する点と、予測によりWフラグを設定する点と、ＬＲＵ方式におけるアクセス順序情報の代わりに使用フラグUを用いる点とが異なる。
【００８３】
＜使用フラグUの説明＞
図１１は、制御部１３８による使用フラグの更新例を示す。同図の上段、中断、下段は、ウェイ０～３に跨るセットＮを構成する４つのキャッシュエントリーを示している。４つのキャッシュエントリー右端の１又は０は、それぞれ使用フラグの値である。この４つの使用フラグUをＵ０～Ｕ３と記す。
【００８４】
同図上段では（Ｕ０～Ｕ３）＝（１、０、１、０）であるので、ウェイ０、２のキャッシュエントリーはアクセスがあったことを、ウェイ１、３のキャッシュエントリーはアクセスがないことを意味する。
【００８５】
この状態で、メモリアクセスがセットＮ内のウェイ１のキャッシュエントリーにヒットした場合、同図中段に示すように、（Ｕ０～Ｕ３）＝（１、１、１、０）に更新される。つまり、実線に示すようにウェイ１の使用フラグＵ１が０から１に更新される。
【００８６】
さらに、同図中段の状態で、メモリアクセスがセットＮ内のウェイ３のキャッシュエン

トリーにヒットした場合、同図下断に示すように、（Ｕ０～Ｕ３）＝（０、０、０、１）に更新される。つまり、実線に示すようにウェイ３の使用フラグＵ１が０から１に更新される。加えて、破線に示すようにウェイ３以外の使用フラグＵ０～Ｕ２が１から０に更新される。これにより、ウェイ３のキャシュエントリーが、ウェイ０～２の各キャッシュエントリーよりも新しくアクセスされたことを意味することになる。

【００８７】

制御部１３８は、キャッシュミス時にＷ＝１のキャッシュエントリーが存在しなければ、使用フラグに基づいてリプレース対象のキャッシュエントリーを決定してリプレースを行う。例えば、制御部１３８は、図５上段では、ウェイ１とウェイ３の何れかをリプレース対象と決定し、図５中段ではウェイ３をリプレース対象と決定し、図５下段ではウェイ０～２の何れかをリプレース対象と決定する。

【００８８】

＜ウィークフラグＷの説明＞

図１２（ａ）ウィークフラグが存在しないと仮定した場合の比較例であり、キャッシュエントリーがリプレースされる様子を示す図である。同図においても、図１１と同様にウェイ０～３に跨るセットＮを構成する４つのキャッシュエントリーを示している。、４つのキャッシュエントリー右端の１又は０は、それぞれ使用フラグの値である。また、データＥのみアクセス頻度の低いデータを、データＡ、Ｂ、Ｃ、Ｄはアクセス頻度の高いデータとする。

【００８９】

同図（ａ）の第１段目の状態で、プロセッサ１がデータＥにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、例えば、Ｕ＝０のキャッシュエントリーの中からアクセス頻度の高いデータＣのキャッシュエントリーがアクセス頻度の低いデータＥにリプレースされ、第２段目の状態となる。

【００９０】

第２段目の状態で、プロセッサ１がデータＣにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、Ｕ＝０のキャッシュエントリーであるアクセス頻度の高いデータＤのキャッシュエントリーがアクセス頻度の高いデータＣにリプレースされ、第３段目の状態となる。

【００９１】

第３段目の状態で、プロセッサ１がデータＤにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、例えば、アクセス頻度の高いデータＣのキャッシュエントリーがアクセス頻度の高いデータＤにリプレースされ、第３段目の状態となる。

【００９２】

同様に、第４段目でも、使用頻度の低いデータＥはリプレース対象として選択されないで、キャッシュメモリーに残っている。

第５段目の状態で、使用頻度の低いデータＥは最も古い（Ｕ＝０）であることから、リプレース対象として選択されて、追い出される。

【００９３】

このように、擬似ＬＲＵ方式において（通常のＬＲＵ方式においても）、アクセス頻度の低いデータＥによって、４ウェイの場合は最悪４回のキャッシュミスを誘発する場合がある。

【００９４】

図１２（ｂ）は、リプレース処理におけるウィークフラグＷの役割を示す説明図である。

同図（ｂ）の第１段目の状態（同図（ａ）の第１段目と同じ）で、プロセッサ１がデータＥにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、例えば、Ｕ＝０のキャッシュエントリーの中からアクセス頻度の高いデータＣのキャッシュエントリーがアクセス頻度の低いデータＥにリプレースされる。このとき、プロセッサ１は、データＥのキャッシュエントリーにウィークフラグＷを１に設定するものとする。これに

より、次のキャッシュミス時にデータＥのキャッシュエントリーが真っ先に追い出され、第２段目の状態となる。

【００９５】

第２段目の状態で、プロセッサ１がデータＣにアクセスすると、キャッシュミスが発生する。このキャッシュミスにより、Ｗ＝１のキャッシュエントリーであるアクセス頻度の低いデータＥのキャッシュエントリーがリプレース対象として選択され、アクセス頻度の高いデータＣにリプレースされ、第３段目の状態となる。

【００９６】

このように、ウィークフラグＷを設けることにより、アクセス頻度の低いデータによるキャッシュミスの誘発を低減することができる。

【００９７】

＜予測処理部の構成＞

図１３は、予測処理部１３９の構成を示すブロック図である。同図の予測処理部１３９は、図３に示した予測処理部３９と比較して、実行部４１３の代わりに実行部４１３ａを備える点が異なっている。実行部４１３ａは、実行部４１３と比べて、Ｃ設定部４１６、Ｗ設定部４１７が追加されている。

【００９８】

Ｃ設定部４１６は、プロセッサのストア命令の進行状況に応じてストアが完了した直前のラインを推定し、推定したラインのＣフラグを１に設定する。

Ｗ設定部４１７は、プロセッサのストア命令の進行状況に応じてストアが完了した直前のラインを推定し、推定したラインのＷフラグを１に設定する。

【００９９】

＜Ｃフラグ設定処理の説明図＞

図１４（ａ）は、Ｃ設定部４１６によるＣフラグ設定処理の説明図である。同図（ａ）において、ラインＮ、Ｎ＋１、・・・、Ｎ＋ｎ、スタートライン、エンドラインは、図６（ａ）と同様である。

【０１００】

Ｃ設定部４１６は、プロセッサ１がシーケンシャルにラインＮ、Ｎ＋１、・・・に対してデータをストアしていく場合に、例えば、ラインＮ＋１に対してストア命令ＳＴ（１）、ＳＴ（２）が実行され、ＳＴ（２）のメモリアクセスアドレスと予測値とが一致した場合に、ラインＮに対するシーケンシャルアクセスによるストアが完了したものと推定し、ラインＮのＣフラグを１に設定する。Ｃフラグが１に設定されたラインＮは、キャッシュミスの発生を待たずにクリーニング処理によりライトバックされる。

【０１０１】

同様、ラインＮ＋２へのストア命令実行中に予測値を一致した場合、ラインＮ＋１に対するシーケンシャルアクセスによるストアが完了したものと推定し、ラインＮ＋１のＣフラグを１に設定する。

【０１０２】

このように、Ｃ設定部４１６は、Ｃフラグの設定をスタートラインからエンドラインまでの範囲内で実行する。

【０１０３】

＜Ｗフラグ設定処理の説明図＞

図１４（ｂ）は、Ｗ設定部４１７によるＷフラグ設定処理の説明図である。同図（ａ）と比べて、Ｃフラグの代わりにＷフラグを設定する点のみが異なり、これ以外は同様であるので説明を省略する。

【０１０４】

Ｗフラグが１に設定されたラインは、キャッシュミスの発生時に真っ先にリプレース対象として選択され、キャッシュメモリから追い出される。

【０１０５】

＜Ｃフラグ設定処理フロー＞

図１５は、Ｃ設定部４１６におけるＣフラグ設定処理の一例を示すフローチャートである。

【０１０６】

同図において、Ｃ設定部４１６は、コマンドレジスタ４０１にＣフラグ設定コマンドが保持されていて（Ｓ４１ｂ）、プロセッサがストア命令を実行したとき（Ｓ４２ｂ）、当該メモリアクセスアドレスと定数とを加算した値を初期値として予測値レジスタ４０８に設定する（Ｓ４３）。この最初のストア命令のメモリアクセスアドレスと予測値レジスタの予測値とは当然一致しない。予測値レジスタ４０８はクリアされているかランダムな値を保持しているからである。

【０１０７】

さらに、プロセッサがストア命令を実行し（Ｓ４４ｂ）、かつメモリアクセスアドレスと予測値レジスタの予測値とが一致する場合には（Ｓ４５）、Ｃ設定部４１６は、直前のラインのラインアドレスを算出し（Ｓ４６ｂ）、算出したラインアドレスがスタートラインからエンドラインまでのアドレス範囲に属していて（Ｓ４７）、直前のラインがキャッシュメモリにエントリーされていれば（Ｓ４８ｂ）、直前のラインのＣフラグを１に設定する（Ｓ４９ｂ）。

【０１０８】

さらに、Ｃ設定部４１６は、直前のラインがエンドラインであれば（Ｓ５０）Ｃフラグ設定処理を終了する。

このように、Ｃ設定部４１６は、プロセッサのストア命令の進行状況に応じて、ストアが完了した直前のラインを推定し、推定したラインのＣフラグを１に設定するので、プロセッサ１では、キャッシュメモリのラインサイズやライン境界を認識する必要がなく、つまり、クリーニングしてもよいラインをラインアドレスとしてプロセッサ１にて管理する必要がない。キャッシュメモリにおいて推定に基づいて効率よくクリーニングを行うことができる。

【０１０９】

＜Ｗフラグ設定処理フロー＞

図１６は、Ｗ設定部４１７におけるＷフラグ設定処理の一例を示すフローチャートである。同図のフローは、図１５のＣフラグ設定処理と比較して、Ｃフラグの代わりにＷフラグを設定する点のみが異なり、これ以外は同様であるので説明を省略する。

【０１１０】

このように、Ｗ設定部４１７は、プロセッサのストア命令の進行状況に応じて、ストアが完了した直前のラインを推定し、推定したラインのＷフラグを１に設定するので、キャッシュメモリにおいて推定に基づいてストアが完了した直前のラインを効率よくリプレース対象とすることができる。その際、プロセッサ１では、キャッシュメモリのラインサイズやライン境界を認識する必要がなく、つまり、リプレースしてもよいラインをラインアドレスとしてプロセッサ１にて管理する必要がない。

【０１１１】

＜クリーニング処理＞

図１７は、制御部１３８におけるクリーニング処理の一例を示すフローチャートである。

【０１１２】

同図のように、制御部１３８は、ループ１の処理（Ｓ９００～Ｓ９１３）において、セットインデックス（ＳＩ）０～１５を順に指定する（Ｓ９０１）ことにより、１６個のすべてのセットに対してループ２の処理を行う。ループ２の処理（Ｓ９００～Ｓ９１３）において、制御部１３８は、セット内のウェイ０～３のＣフラグを読み出す（Ｓ９０３）ことにより、Ｃ＝１のキャッシュエントリーを探索する（Ｓ９０４）。ループ３の処理（Ｓ９０５～９１０）において、制御部１３８は、Ｃ＝１のキャッシュエントリーに対して、サブライン単位のダーティフラグを読み出し（Ｓ９０６）、ダーティであれば（Ｓ９０７）、そのサブランのデータをメモリ２に書き戻し（Ｓ９０８）、当該ダーティフラグを０

にリセットする（Ｓ９０９）。このサブラインデータの書き戻しにおいて、制御部１３８は、ループ４の処理（Ｓ９２０～Ｓ９２３）のように、空きサイクルにおいて（Ｓ９２０）、１ワードずつ書き戻す（Ｓ９２２）

このように、制御部１３８は、全てのキャッシュエントリーのＣフラグを順にチェックして、Ｃ＝１のキャッシュエントリーを探索し、ダーティであればキャッシュメモリ３からメモリ２に書き戻す。

【０１１３】

このように、クリーニング処理では、これ以上書き込みされないことを示すＣフラグを有するキャッシュエントリーを、キャッシュミスが発生する前にライトバックするので、キャッシュミス時にはロードペナルティが発生するだけでライトバックペナルティの発生を低減することができる。これによりキャッシュメモリの効率を向上させ、アクセス速度を向上させることができる。

【０１１４】

＜Ｕフラグ更新処理＞

図１８は、制御部１３８におけるＵフラグ更新処理を示すフローチャートである。同図では、バリッドフラグが０（無効）であるキャッシュエントリーの使用フラグＵは０に初期化されているものとする。

【０１１５】

同図において、制御部１３８は、キャッシュヒットしたとき（ステップＳ６１）、セットインデックスにより選択されたセットにおけるヒットしたウェイの使用フラグＵを１にセットし（ステップＳ６２）、そのセット内の他のウェイの使用フラグＵを読み出し（ステップＳ６３）、読み出した使用フラグＵが全て１であるか否かを判定し（ステップＳ６４）、全て１でなければ終了し、全て１であれば他のウェイの全ての使用フラグＵを０にリセットする（ステップＳ６５）。

【０１１６】

このようにして制御部１３８は、図１１、図１２（ａ）（ｂ）に示した更新例のように、使用フラグＵを更新する。

【０１１７】

＜リプレース処理＞

図１９は、制御部１３８におけるリプレース処理を示すフローチャートである。同図において制御部１３８は、メモリアクセスがミスしたとき（ステップＳ９１）、セットインデックスにより選択されたセットにおける、４つウェイの使用フラグＵ及びウィークフラグＷを読み出し（ステップＳ９２）、Ｗ＝１のウェイが存在するか否かを判定する（ステップＳ９３）。Ｗ＝１のウェイが存在しないと判定された場合、Ｕ＝０のウェイを１つ選択する（ステップＳ９４）。このとき、使用フラグＵが０になっているウェイが複数存在する場合は、制御部１３８はランダムに１つを選択する。また、Ｗ＝１のウェイが存在すると判定された場合、Ｕフラグの値に関わらずＷ＝１のウェイを１つ選択する（ステップＳ９５）。このとき、ウィークフラグＷが１になっているウェイが複数存在する場合は、制御部１３８はランダムに１つを選択する。

【０１１８】

さらに、制御部１３８は、当該セットにおける選択されたウェイのキャッシュエントリーを対象にリプレースし（ステップＳ９６）、リプレース後に当該キャッシュエントリーの使用フラグＵを１に、ウィークフラグＷを０初期化する（ステップＳ９７）。なお、このときバリッドフラグＶ、ダーティフラグＤは、それぞれ１、０に初期化される。

【０１１９】

このように、Ｗ＝１のウェイが存在しない場合、リプレース対象は、使用フラグＵが０のキャッシュエントリーの中から１つ選択される。

また、Ｗ＝１のウェイが存在する場合、リプレース対象は、使用フラグＵが０であると１であるとを問わず、Ｗ＝１のウェイのキャッシュエントリーから１つ選択される。これにより図１４（ａ）（ｂ）に示したように、アクセス頻度の低いデータがキャッシュメモ

リに残ることによるキャッシュミスの誘発を低減することができる。
【０１２０】
以上説明してきたように、本実施の形態におけるキャッシュメモリによれば、プロセッサ１によるストア命令の進行状態から、ストアが完了してラインを推定し、推定したラインに対してＣフラグ又はＷフラグを設定するので、ストアが完了した直前のラインを効率よくリプレース対象を指定することができる。その際、プロセッサ１によってキャッシュメモリのライン境界やラインサイズを管理する必要がなく、キャッシュ管理のための負荷を小さくすることができる。
【０１２１】
また、ウィークフラグＷ＝１でかつダーティフラグ＝１のラインを、プロセッサからこれ以上書き込みがなされないラインとして、クリーニングすることにより、キャッシュミス時のライトバックペナルティを低減することができる。
【０１２２】
また、これ以上使用されないキャッシュエントリーにＷ＝１が設定され、Ｗ＝１のキャッシュエントリーが真っ先にリプレース対象として選択されるので、アクセス頻度の低いデータがキャッシュメモリに残ることによるキャッシュミスの誘発を低減することができる。
【０１２３】
また、従来のＬＲＵ方式におけるアクセス順序を示すデータの代わりに１ビットの使用フラグを用いる擬似ＬＲＵ方式を採用することにより、アクセス順序データとして１ビットのフラグでよいので、アクセス順序データのデータ量が少ないこと及び更新が簡単であることからハードウェア規模を小さくすることができる。
【０１２４】
＜変形例＞
（１）図４（ａ）（ｂ）（ｃ）に示した各命令は、コンパイラによりプログラム中に挿入してもよい。その際、コンパイラは、例えば配列データの書き込みや、圧縮動画データをデコードする際のブロックデータの書き込み等、これ以上書き込みをしないプログラム位置に、上記各命令を挿入するようにすればよい。
（２）上記クリーニング処理において無効化処理を行う構成としてもよい。すなわち、図１７に示したフローチャートにおいて、Ｓ９０７にてダーティでないと判定された場合、さらに、当該キャッシュエントリーを無効化（Ｖフラグをリセット）するステップを追加してもよい。さらに、クリーニング処理におけるライトバックの後に無効化するステップを追加してもよい。
【産業上の利用可能性】
【０１２５】
本発明は、メモリアクセスを高速化するためのキャッシュメモリに適しており、例えば、オンチップキャッシュメモリ、オフチップキャッシュメモリ、データキャッシュメモリ、命令キャッシュメモリ等に適している。
【図面の簡単な説明】
【０１２６】
【図１】本発明の実施の形態１におけるプロセッサ、キャッシュメモリ、メモリを含むシステムの概略構成を示すブロック図である。
【図２】キャッシュメモリの構成例を示すブロック図である。
【図３】予測処理部の構成例を示すブロック図である。
【図４】（ａ）スタートアドレスレジスタにスタートアドレスを書き込む命令の一例を示す。（ｂ）サイズレジスタにサイズを書き込む命令の一例を示す。（ｃ）コマンドレジスタにコマンドを書き込む命令の一例を示す。（ｄ）コマンドの一例を示す。
【図５】スタートアライナ及びエンドアライナの説明図である。
【図６】（ａ）プリフェッチ部による予測プリフェッチの説明図である。（ｂ）タッチ部による予測タッチの説明図である。

【図７】プリフェッチ部における予測プリフェッチ処理の一例を示すフローチャートである。
【図８】タッチ部における予測タッチ処理の一例を示すフローチャートである。
【図９】本発明の実施の形態２におけるキャッシュメモリの構成を示すブロック図である。
【図１０】キャッシュエントリーのビット構成を示す。
【図１１】制御部による使用フラグの更新例を示す。
【図１２】（ａ）ウィークフラグが存在しない場合にキャッシュエントリーがリプレースされる様子を示す図である。（ｂ）リプレース処理におけるウィークフラグＷの役割を示す説明図である。
【図１３】予測処理部の構成を示すブロック図である。
【図１４】（ａ）Ｃ設定部によるＣフラグ設定処理の説明図である。（ｂ）Ｗ設定部によるＷフラグ設定処理の説明図である。
【図１５】Ｃフラグ設定部におけるＣフラグ設定処理の一例を示すフローチャートである。
【図１６】Ｗフラグ設定部におけるＷフラグ設定処理の一例を示すフローチャートである。
【図１７】制御部におけるクリーニング処理の一例を示すフローチャートである。
【図１８】制御部におけるＵフラグ更新処理を示すフローチャートである。
【図１９】制御部におけるリプレース処理を示すフローチャートである。

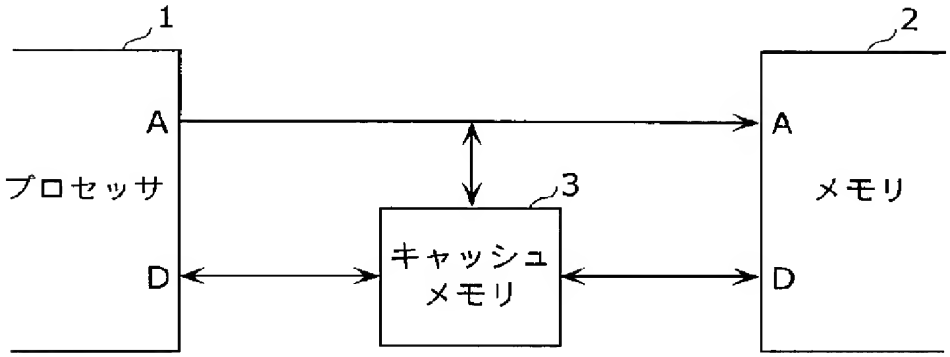
【符号の説明】

【０１２７】

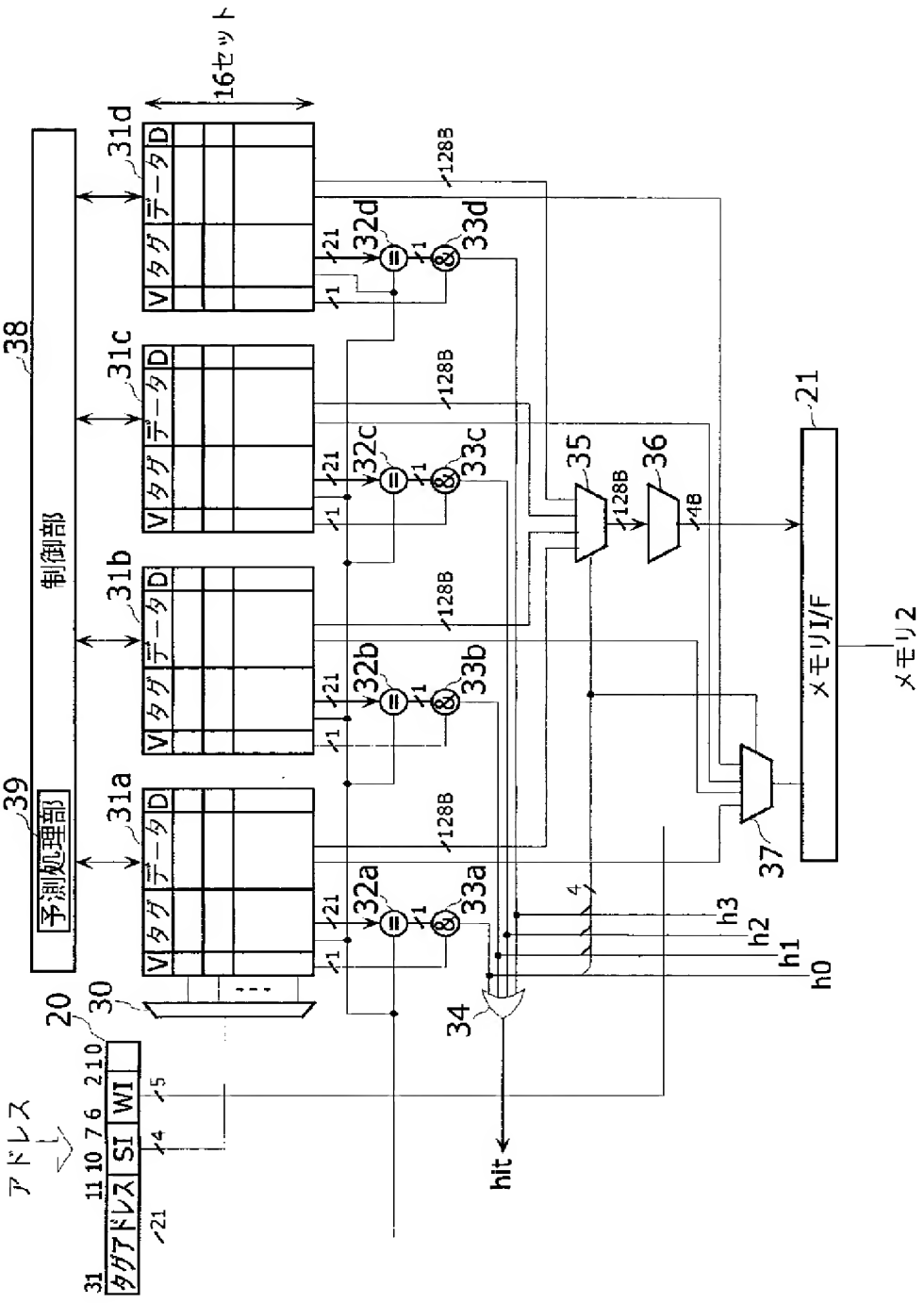
１	プロセッサ
２	メモリ
３	キャッシュメモリ
２０	アドレスレジスタ
２１	メモリＩ／Ｆ
３０	デコーダ
３１ａ～３１ｄ	ウェイ
３２ａ～３２ｄ	比較器
３３ａ～３３ｄ	アンド回路
３４	オア回路
３５	セレクタ
３６	セレクタ
３７	デマルチプレクサ
３８	制御部
３９	予測処理部
１３１ａ	ウェイ
１３１ａ～１３１ｄ	ウェイ
１３１ｂ～１３１ｄ	ウェイ
１３８	制御部
１３９	予測処理部
４０１	コマンドレジスタ
４０２	スタートアドレスレジスタ
４０３	サイズレジスタ
４０４	加算器
４０５ａ、４０５ｂ	スタートアライナ
４０６ａ、４０６ｂ	エンドアライナ
４０７	アクセスアドレスレジスタ
４０８	予測値レジスタ

４０９　　定数レジスタ
４１０　　セレクタ
４１１　　加算器
４１２　　比較器
４１３　　実行部
４１３ａ　実行部
４１４　　プリフェッチ部
４１５　　タッチ部
４１６　　Ｃ設定部
４１７　　Ｗ設定部

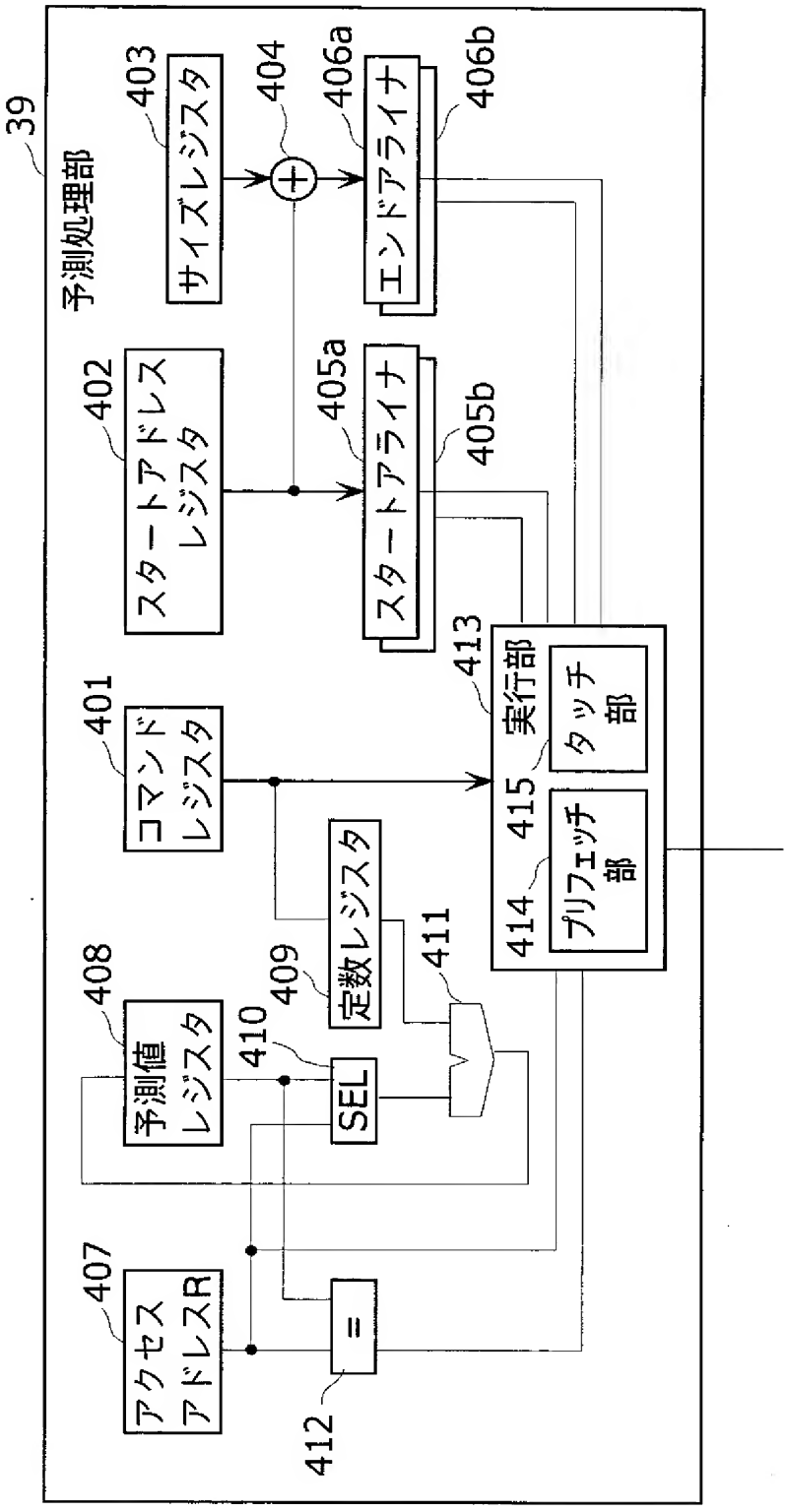
【書類名】図面
【図１】



【図２】

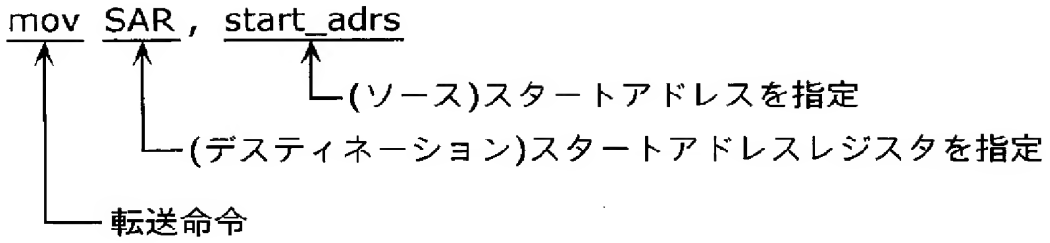

16セット
31d
31c
31b
31a
V
タグ
データ
D
128B
32d
32c
32b
32a
33d
33c
33b
33a
21
1
35
36
128B
4B
37
21
メモリI/F
メモリ2
38
制御部
39
予測処理部
20
30
34
h3
h2
h1
h0
hit
アドレス
31
タグアドレス
SI
WI
11 10 7 6 2 1 0
5
4

【図３】

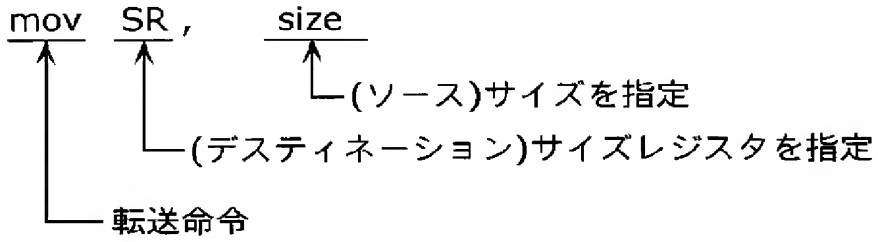

39
予測処理部
403
サイズレジスタ
404
406a
エンドアライナ
406b
402
スタートアドレス
レジスタ
405a
スタートアライナ
405b
401
コマンド
レジスタ
413
実行部
タッチ
部
415
プリフェッチ
部
414
408
予測値
レジスタ
409
定数レジスタ
410
SEL
411
407
アクセス
アドレスR
=
412

【図４】

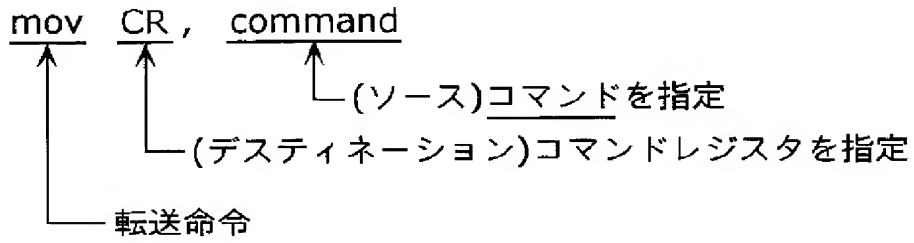
(a)


mov SAR , start_adrs
(ソース)スタートアドレスを指定
(デスティネーション)スタートアドレスレジスタを指定
転送命令

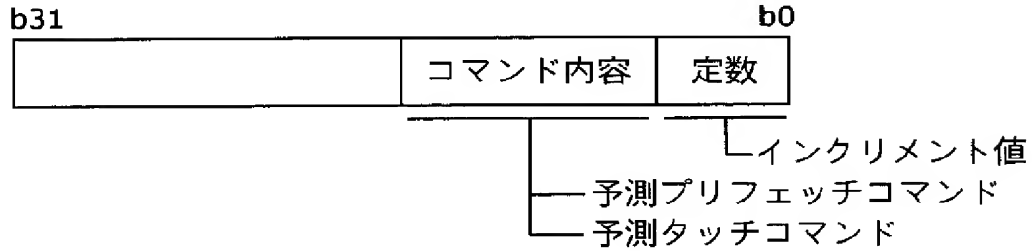
(b)


mov SR , size
(ソース)サイズを指定
(デスティネーション)サイズレジスタを指定
転送命令

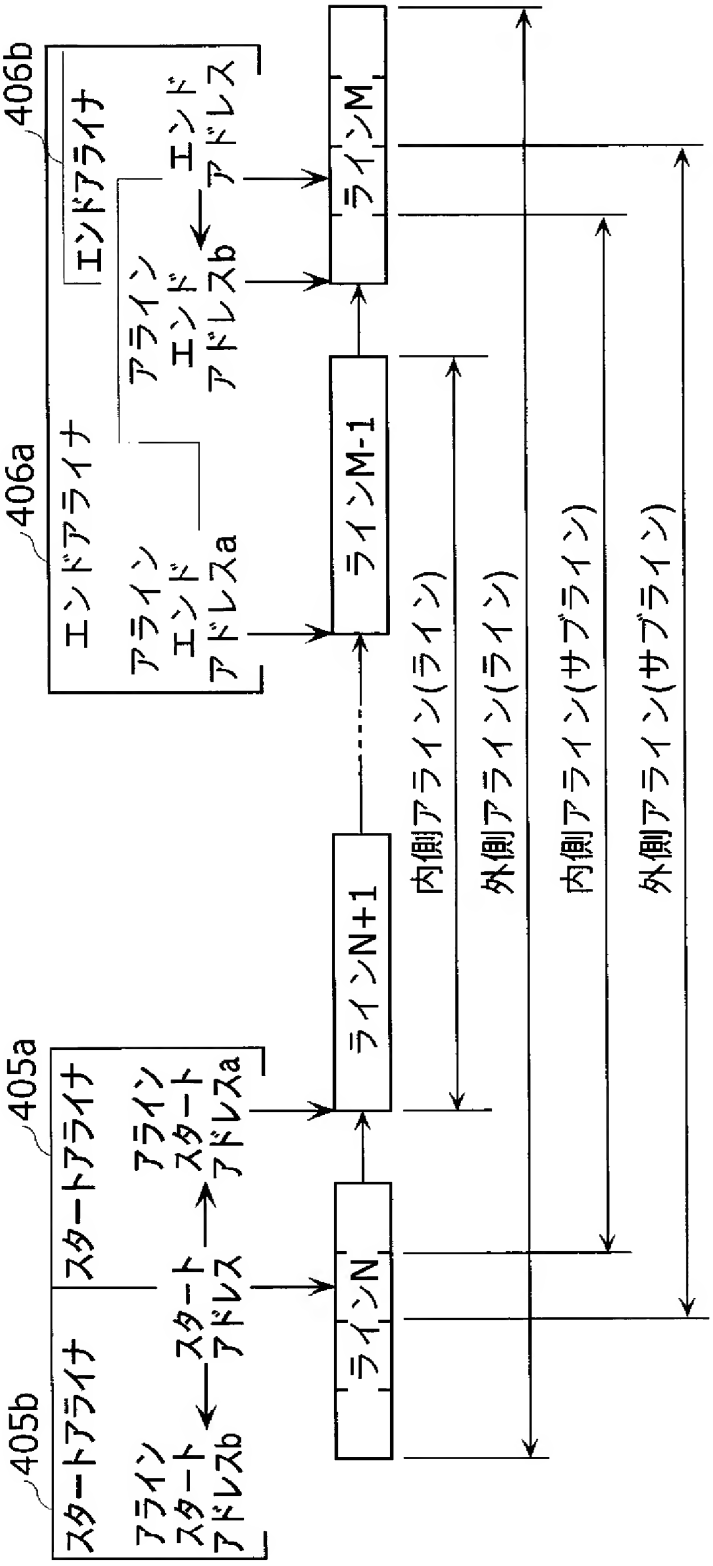
(c)


mov CR , command
(ソース)コマンドを指定
(デスティネーション)コマンドレジスタを指定
転送命令

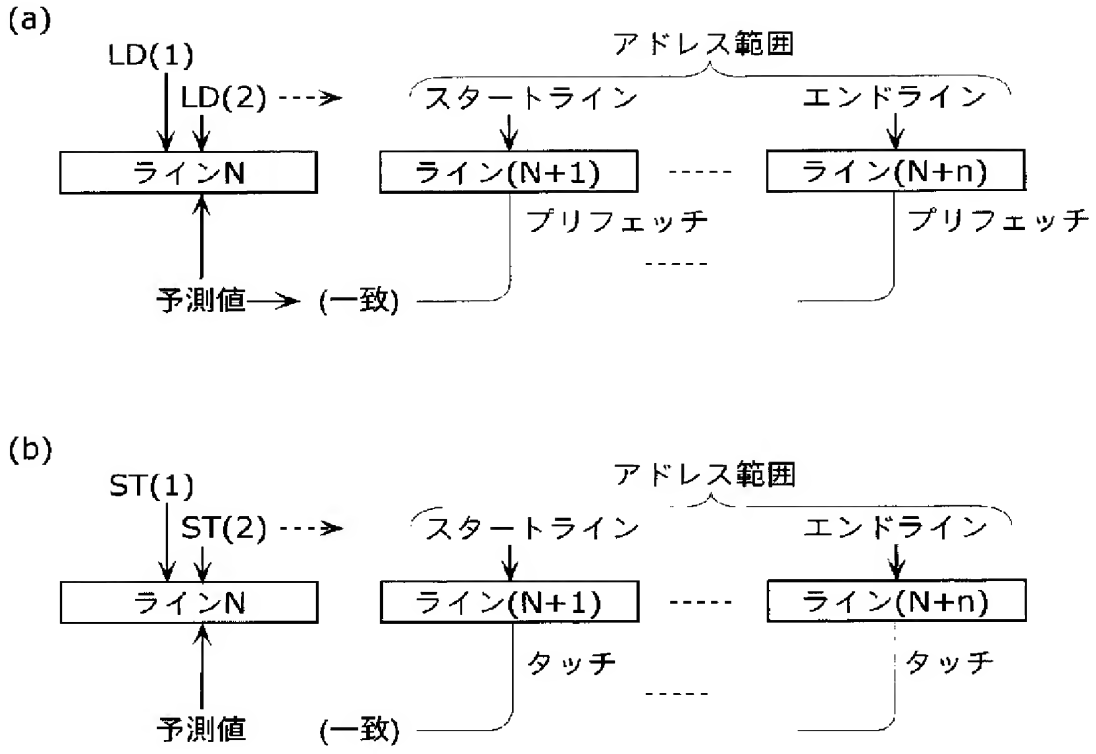
(d)


b31
b0
コマンド内容
定数
インクリメント値
予測プリフェッチコマンド
予測タッチコマンド

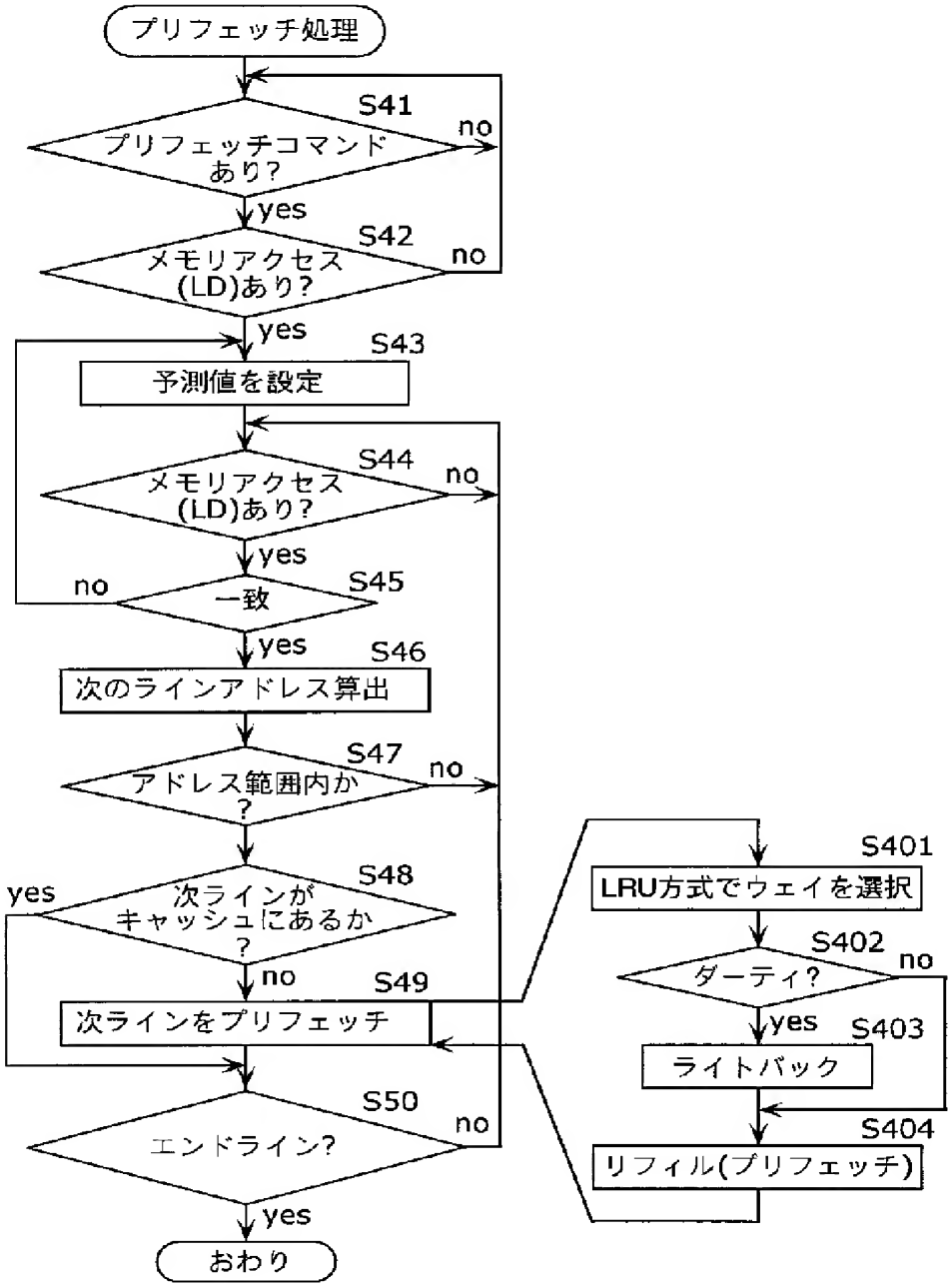
【図５】


405b
スタートアライナ
アライン
スタート
アドレスb
スタート
アドレス
405a
スタートアライナ
アライン
スタート
アドレスa
406a
エンドアライナ
アライン
エンド
アドレスa
406b
エンドアライナ
アライン
エンド
アドレスb
エンド
アドレス
ラインN
ラインN+1
ラインM-1
ラインM
内側アライン(ライン)
外側アライン(ライン)
内側アライン(サブライン)
外側アライン(サブライン)

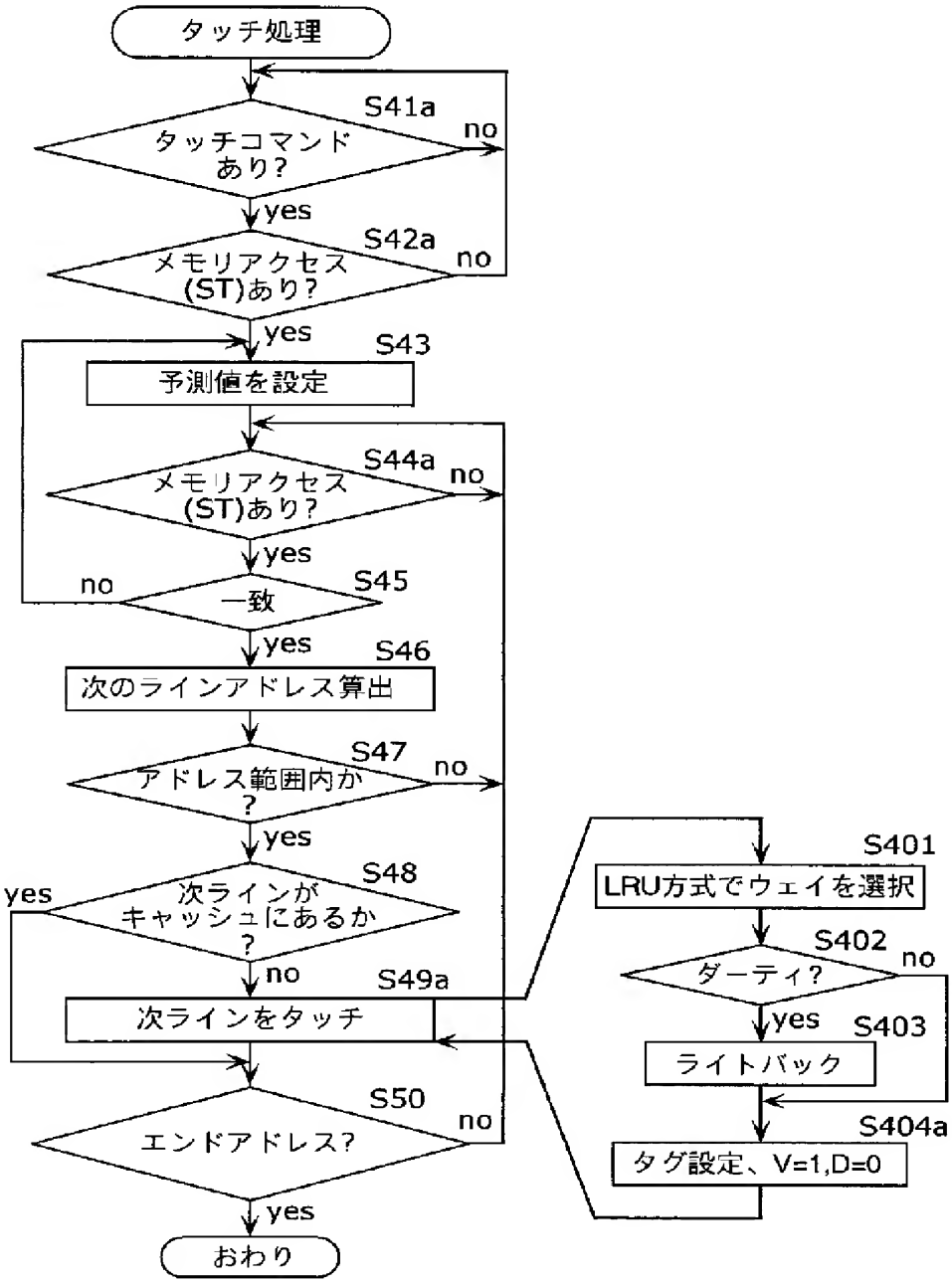
【図６】


(a)
LD(1)
LD(2)
アドレス範囲
スタートライン
エンドライン
ラインN
ライン(N+1)
ライン(N+n)
プリフェッチ
プリフェッチ
予測値
(一致)
(b)
ST(1)
ST(2)
アドレス範囲
スタートライン
エンドライン
ラインN
ライン(N+1)
ライン(N+n)
タッチ
タッチ
予測値
(一致)

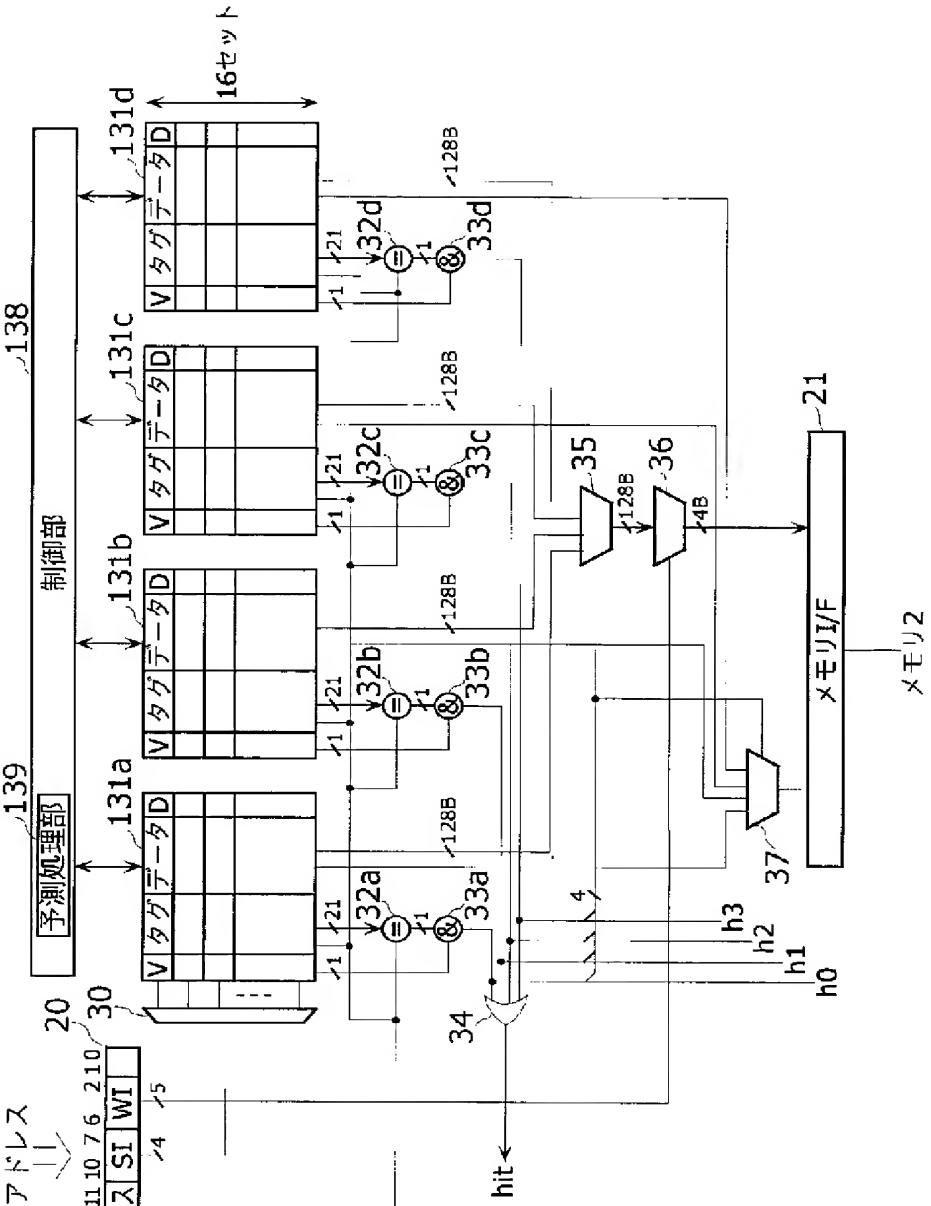
【図７】


プリフェッチ処理
S41
プリフェッチコマンドあり?
no
yes
S42
メモリアクセス(LD)あり?
no
yes
S43
予測値を設定
S44
メモリアクセス(LD)あり?
no
yes
no
S45
一致
yes
S46
次のラインアドレス算出
S47
アドレス範囲内か?
no
S48
yes
次ラインがキャッシュにあるか?
no
S49
次ラインをプリフェッチ
S50
エンドライン?
no
yes
おわり
S401
LRU方式でウェイを選択
S402
ダーティ?
no
yes
S403
ライトバック
S404
リフィル(プリフェッチ)

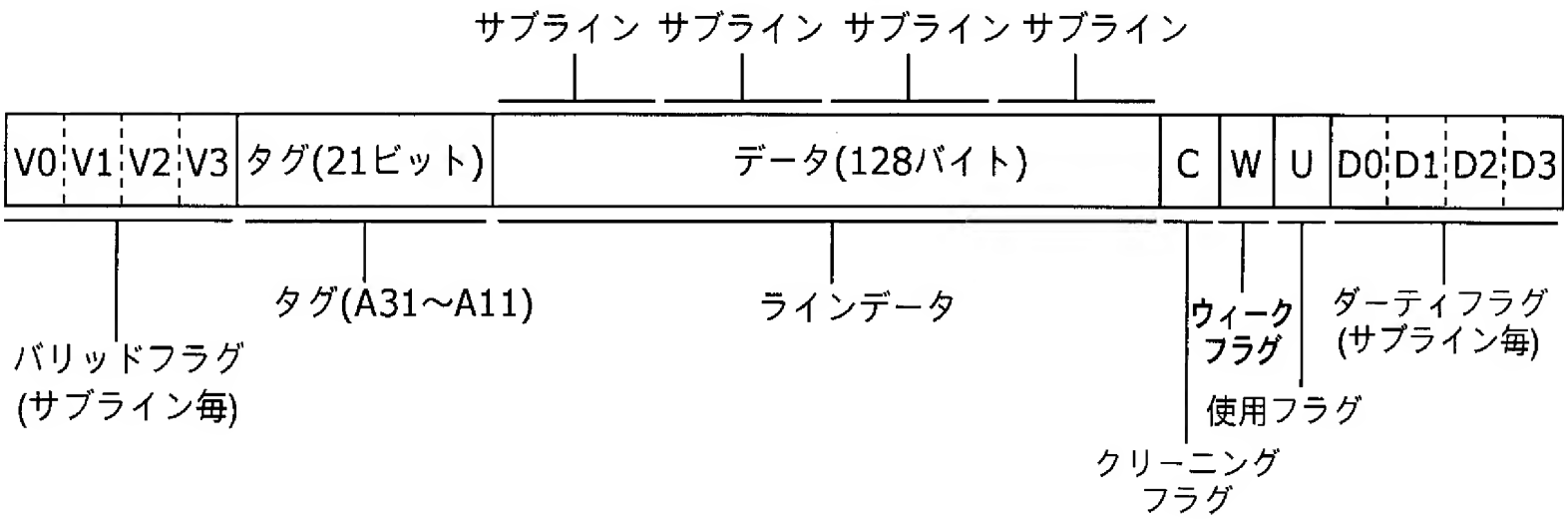
【図８】


タッチ処理
S41a
タッチコマンド
あり?
no
yes
S42a
メモリアクセス
(ST)あり?
no
yes
S43
予測値を設定
S44a
メモリアクセス
(ST)あり?
no
yes
no
S45
一致
yes
S46
次のラインアドレス算出
S47
アドレス範囲内か
?
no
yes
S48
次ラインが
キャッシュにあるか
?
yes
no
S49a
次ラインをタッチ
S50
エンドアドレス?
no
yes
おわり
S401
LRU方式でウェイを選択
S402
ダーティ?
no
yes
S403
ライトバック
S404a
タグ設定、V=1,D=0

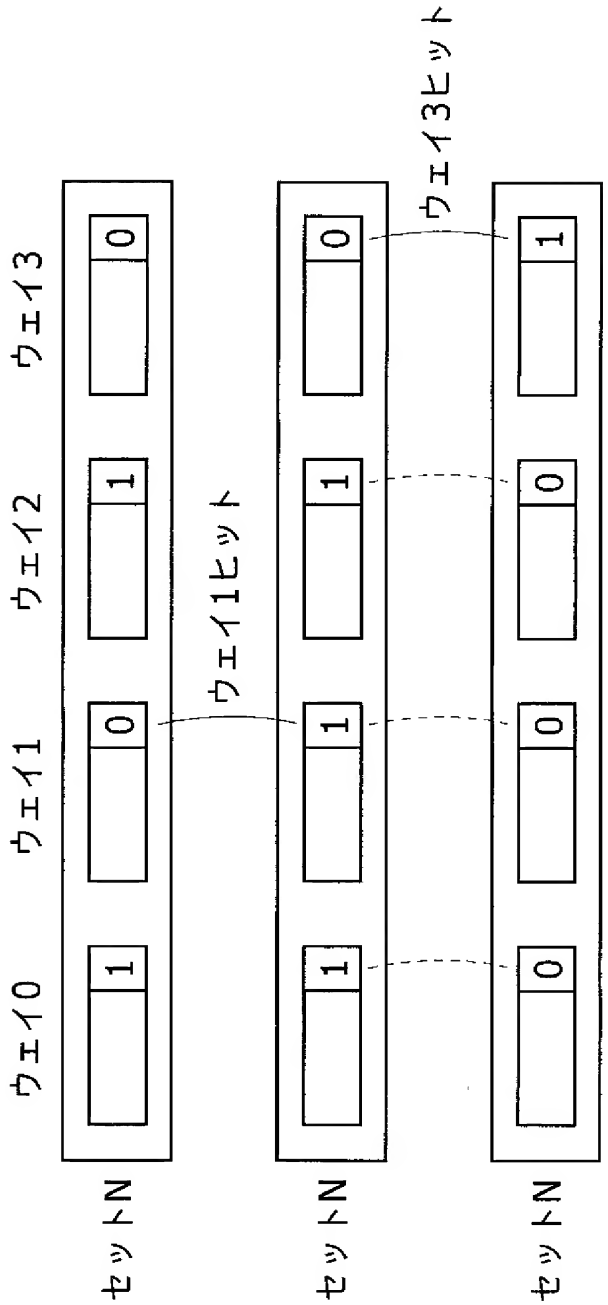
【図９】


16セット
131d
131c
131b
131a
V タグ データ D
138
制御部
139
予測処理部
20
30
アドレス
31 11 10 7 6 2 1 0
タグアドレス SI WI
21
5
4
32a
32b
32c
32d
33a
33b
33c
33d
34
35
36
37
128B
4B
4
h3
h2
h1
h0
hit
メモリI/F
メモリ2

【図１０】

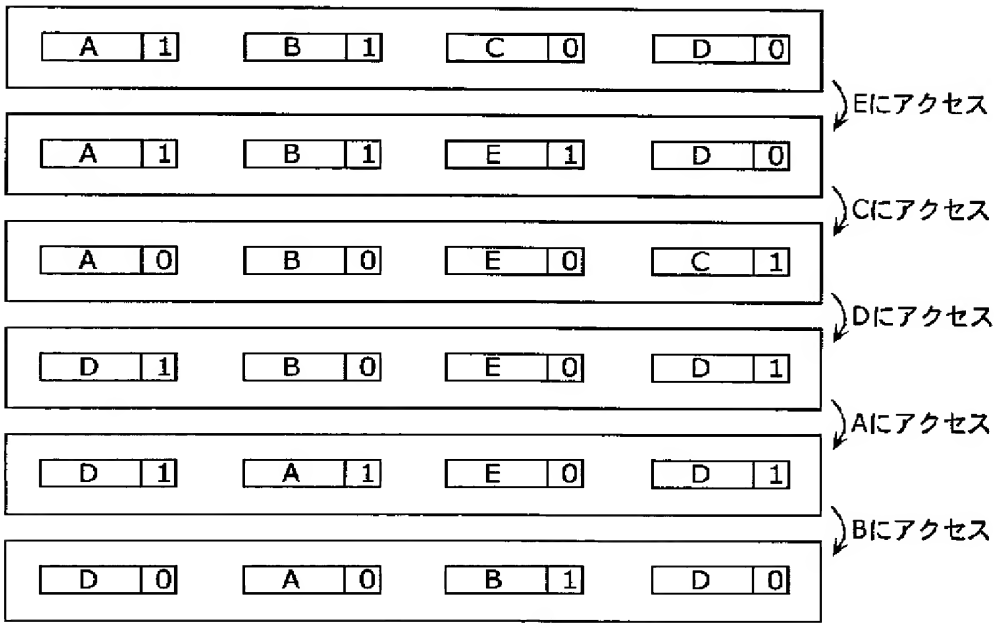

サブライン
サブライン
サブライン
サブライン
V0
V1
V2
V3
タグ(21ビット)
データ(128バイト)
C
W
U
D0
D1
D2
D3
バリッドフラグ
(サブライン毎)
タグ(A31～A11)
ラインデータ
ウィーク
フラグ
ダーティフラグ
(サブライン毎)
使用フラグ
クリーニング
フラグ

【図１１】

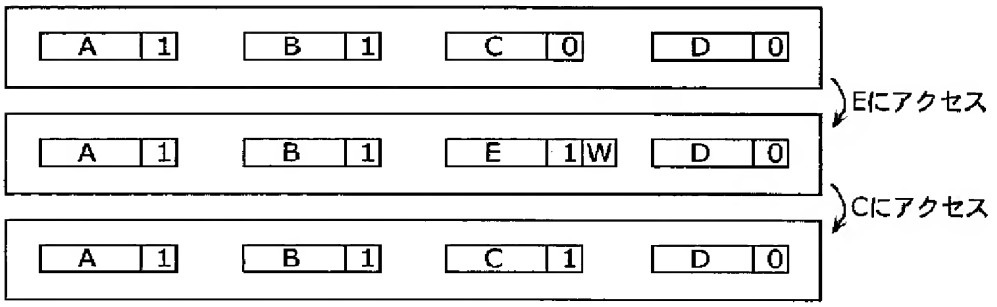

ウェイ0
ウェイ1
ウェイ2
ウェイ3
1
0
1
0
ウェイ1ビット
ウェイ3ビット
1
0
1
0
1
0
1
0
セットN
セットN
セットN

【図１２】

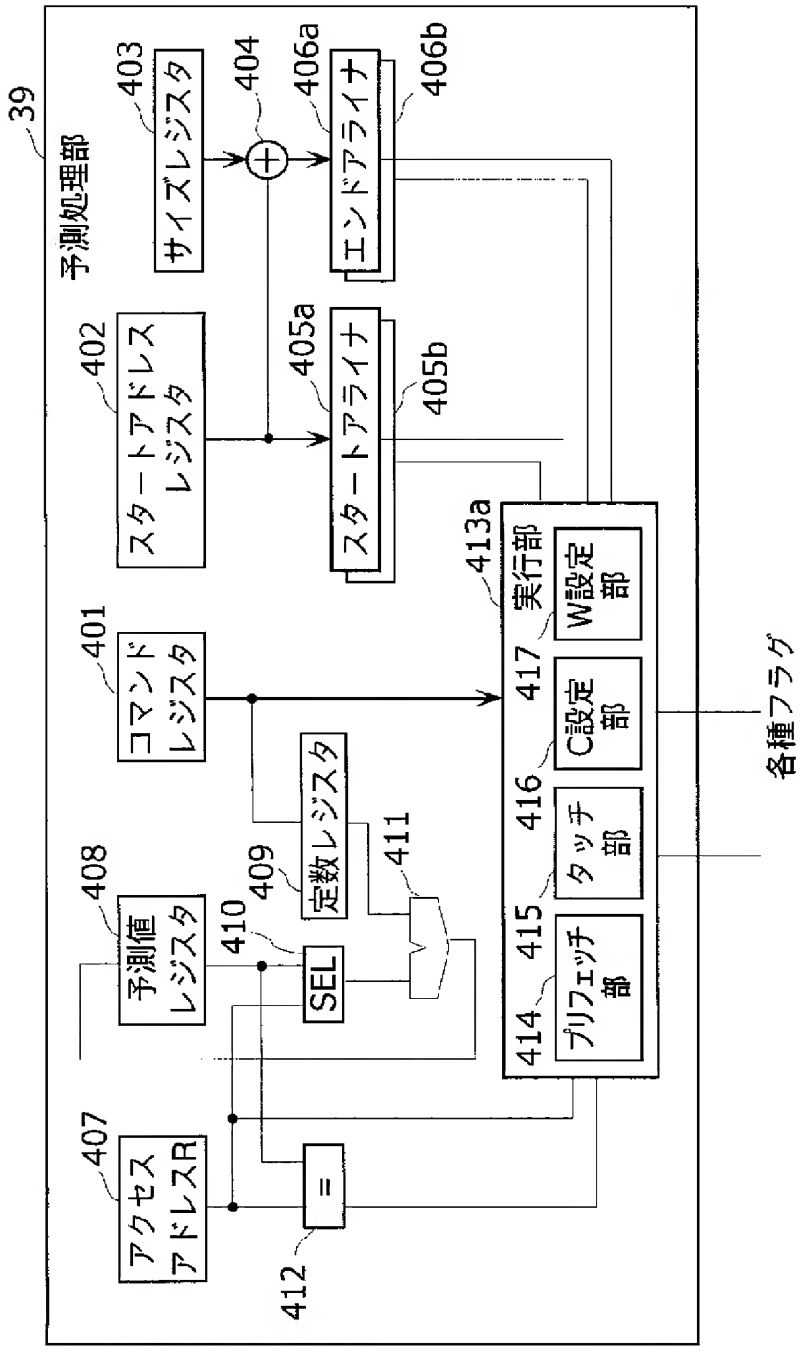
(a)


A 1 B 1 C 0 D 0
Eにアクセス
A 1 B 1 E 1 D 0
Cにアクセス
A 0 B 0 E 0 C 1
Dにアクセス
D 1 B 0 E 0 D 1
Aにアクセス
D 1 A 1 E 0 D 1
Bにアクセス
D 0 A 0 B 1 D 0

(b)

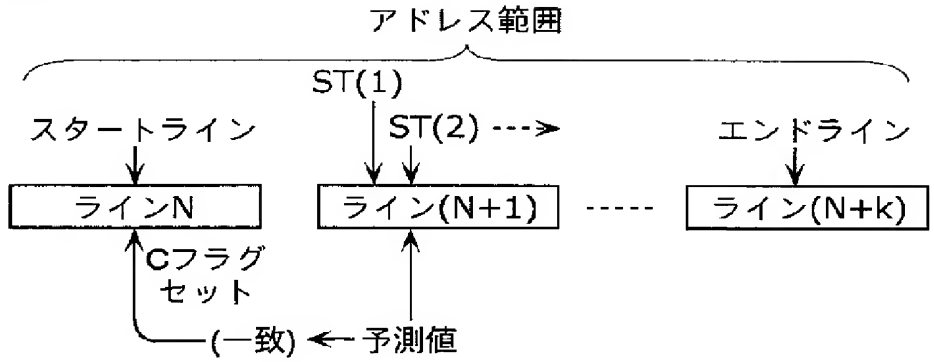

A 1 B 1 C 0 D 0
Eにアクセス
A 1 B 1 E 1 W D 0
Cにアクセス
A 1 B 1 C 1 D 0

【図１３】

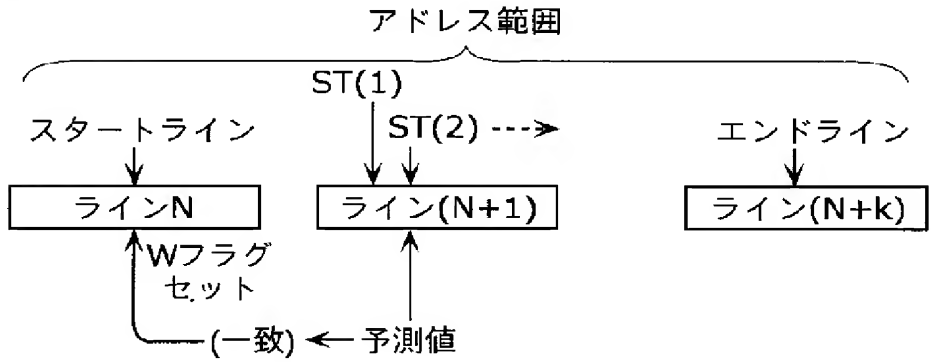

39
予測処理部
401
コマンドレジスタ
402
スタートアドレスレジスタ
403
サイズレジスタ
404
405a
スタートアライナ
405b
406a
エンドアライナ
406b
407
アクセスアドレスR
408
予測値レジスタ
409
定数レジスタ
410
SEL
411
412
=
413a
実行部
414
プリフェッチ部
415
416
タッチ部
417
C設定部
W設定部
各種フラグ

【図１４】

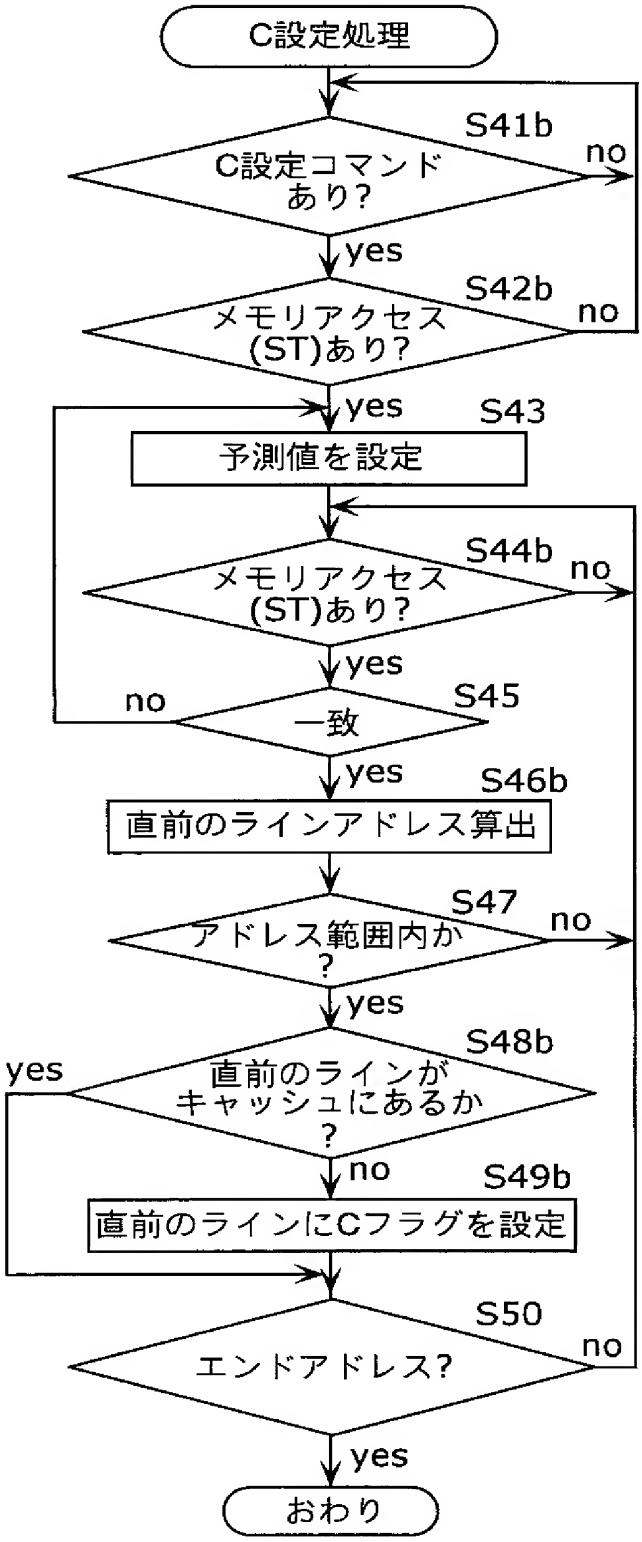
(a)


アドレス範囲
ST(1)
スタートライン
ST(2)
エンドライン
ラインN
ライン(N+1)
ライン(N+k)
Cフラグ
セット
(一致)
予測値

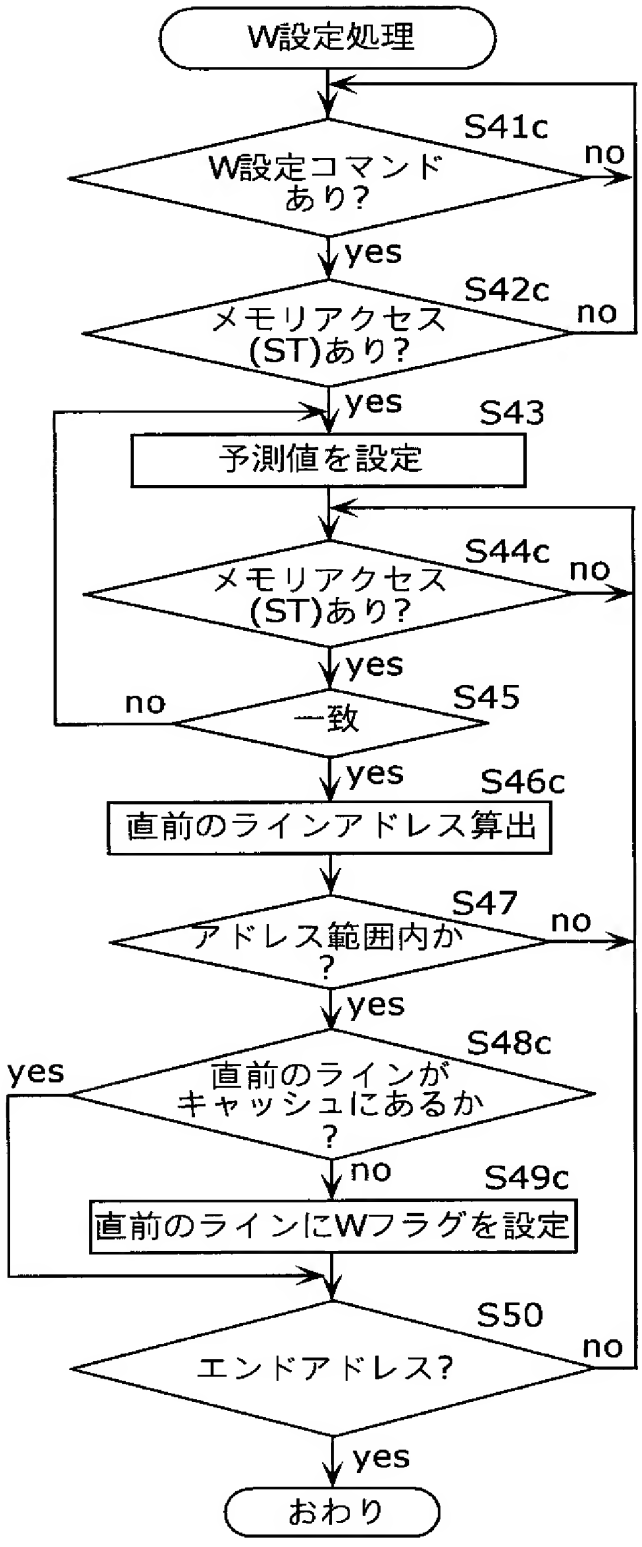
(b)


アドレス範囲
ST(1)
スタートライン
ST(2)
エンドライン
ラインN
ライン(N+1)
ライン(N+k)
Wフラグ
セット
(一致)
予測値

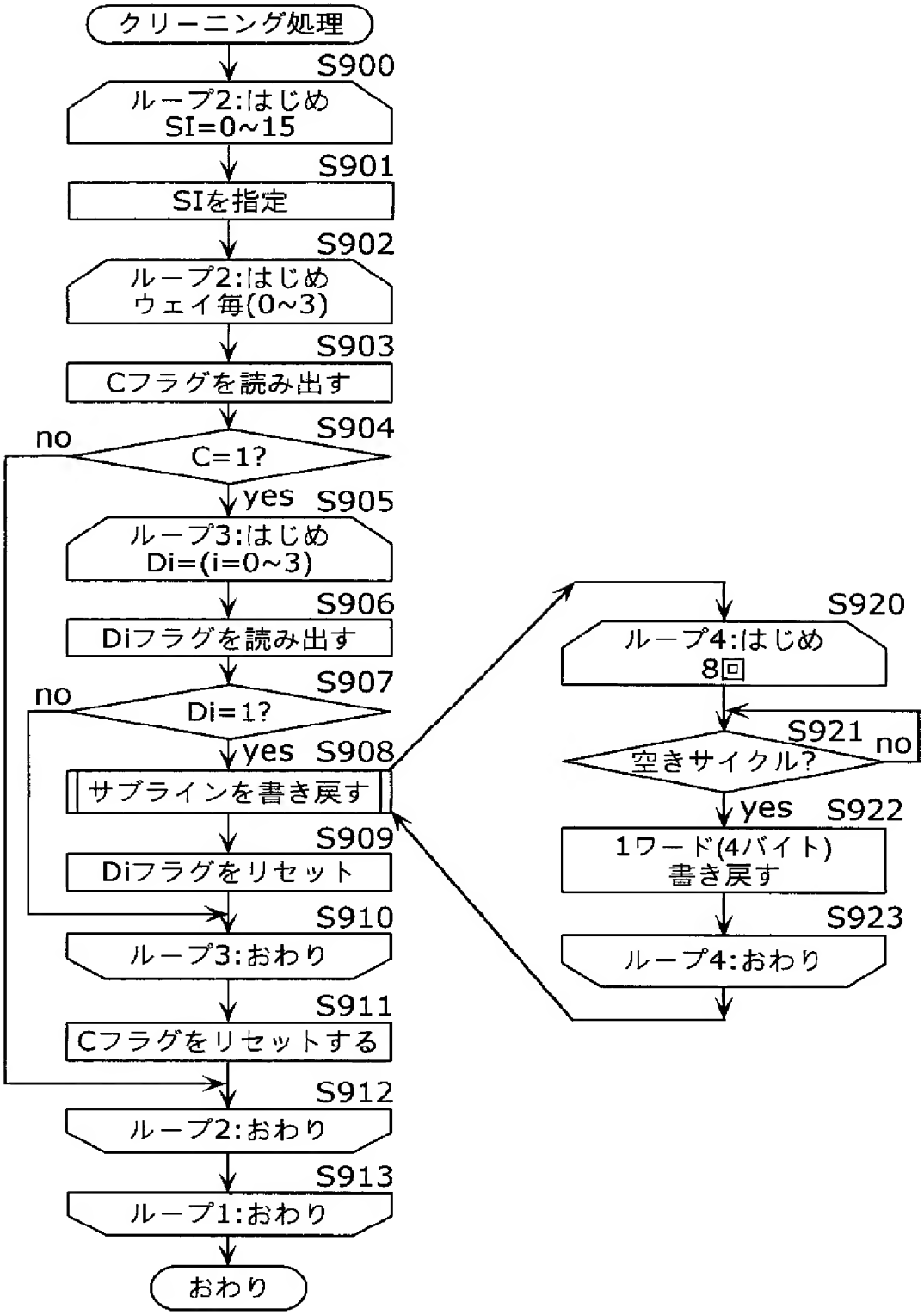
【図１５】


C設定処理
S41b
C設定コマンドあり?
no
yes
S42b
メモリアクセス(ST)あり?
no
yes
S43
予測値を設定
S44b
メモリアクセス(ST)あり?
no
yes
no
S45
一致
yes
S46b
直前のラインアドレス算出
S47
アドレス範囲内か?
no
yes
S48b
直前のラインがキャッシュにあるか?
yes
no
S49b
直前のラインにCフラグを設定
S50
エンドアドレス?
no
yes
おわり

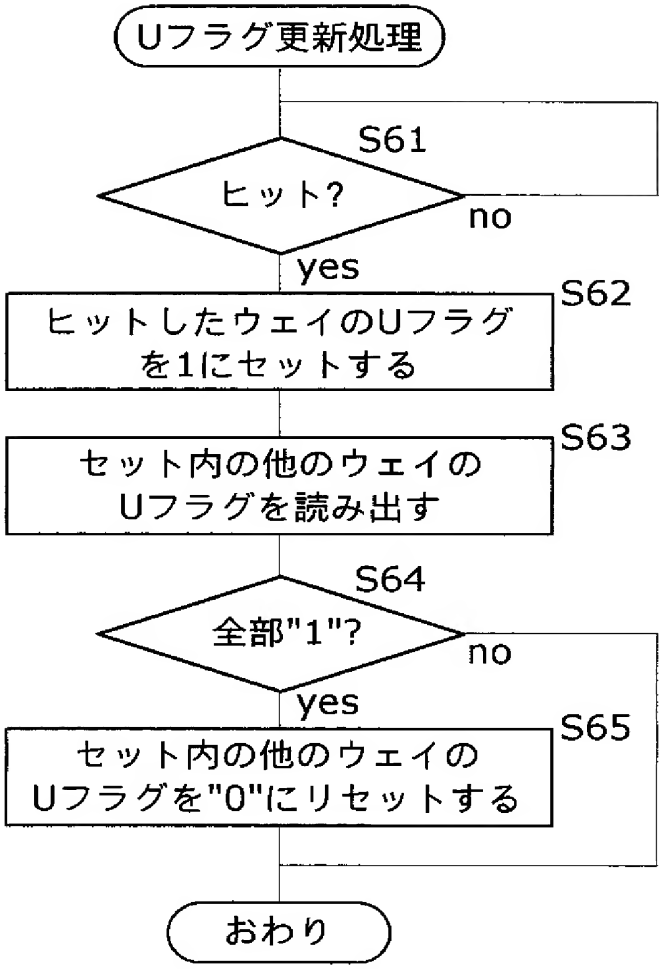
【図１６】


W設定処理
S41c
W設定コマンドあり?
no
yes
S42c
メモリアクセス(ST)あり?
no
yes
S43
予測値を設定
S44c
メモリアクセス(ST)あり?
no
yes
no
S45
一致
yes
S46c
直前のラインアドレス算出
S47
アドレス範囲内か?
no
yes
S48c
直前のラインがキャッシュにあるか?
yes
no
S49c
直前のラインにWフラグを設定
S50
エンドアドレス?
no
yes
おわり

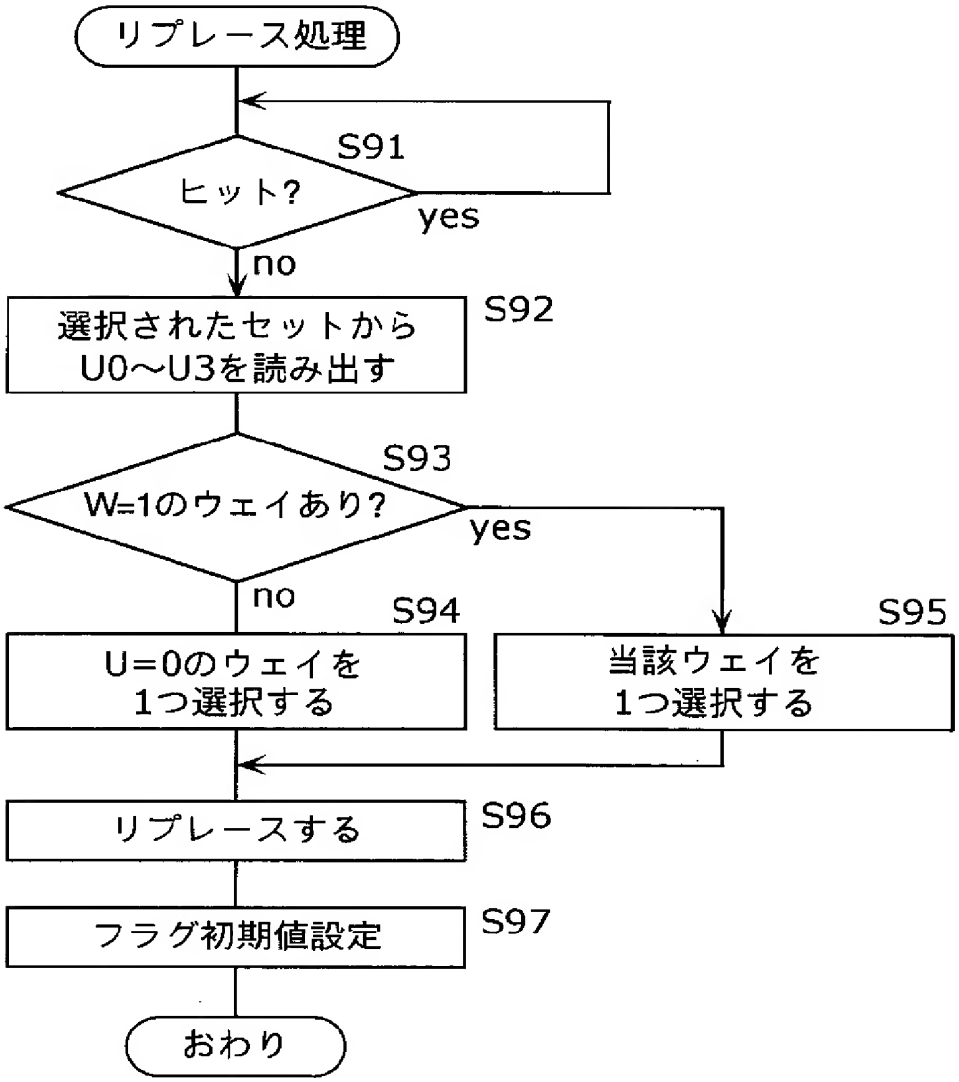
【図１７】


クリーニング処理
S900
ループ2:はじめ
SI=0~15
S901
SIを指定
S902
ループ2:はじめ
ウェイ毎(0~3)
S903
Cフラグを読み出す
S904
C=1?
no
yes
S905
ループ3:はじめ
Di=(i=0~3)
S906
Diフラグを読み出す
S907
Di=1?
no
yes
S908
サブラインを書き戻す
S909
Diフラグをリセット
S910
ループ3:おわり
S911
Cフラグをリセットする
S912
ループ2:おわり
S913
ループ1:おわり
おわり
S920
ループ4:はじめ
8回
S921
空きサイクル?
no
yes
S922
1ワード(4バイト)
書き戻す
S923
ループ4:おわり

【図１８】


Uフラグ更新処理
S61
ヒット?
no
yes
S62
ヒットしたウェイのUフラグ
を1にセットする
S63
セット内の他のウェイの
Uフラグを読み出す
S64
全部"1"?
no
yes
S65
セット内の他のウェイの
Uフラグを"0"にリセットする
おわり

【図１９】


リプレース処理
S91
ヒット?
yes
no
選択されたセットから
U0～U3を読み出す
S92
S93
W=1のウェイあり?
yes
no
S94
U=0のウェイを
1つ選択する
S95
当該ウェイを
1つ選択する
リプレースする
S96
フラグ初期値設定
S97
おわり

【書類名】要約書
【要約】
【課題】　ソフトウェア的な性能劣化を招くことなく、プロセッサの動作状況を監視することにより同期を取りつつ適切なタイミングでキャッシュを操作するキャッシュメモリシステムを提供する。
【解決手段】プロセッサから出力されるメモリアクセスの進行状況に基づいて次にプリフェッチすべきラインアドレスを予測する予測処理部３９を有し、予測処理部３９は、メモリからキャッシュメモリに予測されたラインアドレスのデータをプリフェッチするプリフェッチ部４１４と、メモリからキャッシュメモリにデータをロードすることなく、予測されたラインアドレスをタグとしてキャッシュエントリーに設定し、バリッドフラグを有効にするタッチ部４１５とを備える。
【選択図】　図３

出願人履歴

０００００５８２１
19900828
新規登録

大阪府門真市大字門真１００６番地
松下電器産業株式会社